IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 11, 2021, accepted December 29, 2021, date of publication January 11, 2022, date of current version January 21, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3141767

Collaborative Forecasting and Analysis of Fish Catch in Hokkaido From Multiple Scales by Using Neural Network and ARIMA Model

YUE ZHANG, MASATO YAMAMOTO, GENKI SUZUKI, (Member, IEEE), AND HIROYUKI SHIOYA

Division of Information and Electronic Engineering, Muroran Institute of Technology, Muroran 050-8585, Japan

Corresponding author: Yue Zhang (20096007@mmm.muroran-it.ac.jp)

This research was supported by the Strategic Information and Communications Research and Development Promotion Programme (SCOPE) through the Ministry of Internal Affairs and Communications under Grant 16771288.

ABSTRACT Fishery catch forecasting is a crucial aspect of aquatic research because of its relevance to establishing effective fishery management and resource allocation systems. In this study, we aim to forecast and analyze fish catch by collaboratively processing data using methods at multiple scales. To this end, we propose two computational fishery catch forecasting models. A neural network model based on the multi-timescale features of a convolutional neural network and a long short-term memory neural network is proposed and implemented to forecast short-term measures for the daily catch in the eastern ports of Hokkaido, Japan. Similarly, we propose a long-term catch forecasting and analysis model combining the autoregressive integrated moving average (ARIMA) method and a neural network to explore short-term water temperature and long-term catch dependence in the case of sparse data; we implement this method to investigate the total monthly catch in Hokkaido. The experimental results demonstrate that the proposed methods were able to effectively forecast and analyze fishery catch based on different data scales, volumes, and other complex situations. This is also the first work in the field that considers multiple perspectives.

INDEX TERMS ARIMA model, fishery catch forecasting, neural network.

I. INTRODUCTION

Coastal fishing is an important primary industry in Japan. Hokkaido, an island prefecture in northern Japan, is the largest fishing region in the country, accounting for approximately one-quarter of the coastal fishery products of Japan [1]. However, after peaking in 1984 at 1.282 billion tons, Japanese fishery and aquaculture production declined rapidly and, although this downward trend leveled off after 1995, it has continued to the present day [2]. In this context, accurate catch forecasts can help fishery operators make decisions and perform efficiently. In particular, fishermen and ports can effectively arrange fishing work, and relevant industrial chains such as logistics can be supported to improve logistical efficiency and ensure the freshness of aquatic products [3]. This can greatly relieve pressure on fishery workers. Therefore, catch forecasting is among the most important tasks in the fishery industry [4].

Research on fishery catch forecasting has considered both long-term catch forecasting on a scale of months or years and short-term forecasting on a scale of days. For long-term catch projections, Komiya studied annual catch changes in spiny lobsters and forecasting issues [5]. Leathwick *et al.* used data on water temperature and salt concentrations with a decision tree to forecast snapper catches and fishing probabilities off the coast over short periods of time [6]. Moreover, Kokaki *et al.* developed a fish catch forecasting method using a state-space model that described the probabilistic behavior of fish inside nets [7]. Long-term catch forecasting can reflect macrolevel trend changes in the catch, whereas short-term catch forecasting can reflect specific changes in the catch in detail. Therefore, a combination of long-term and short-term forecasts can comprehensively reflect changes in fish catches, and the two approaches can confirm and complement each other to simulate and summarize the patterns to the greatest extent. Particularly, if changes over different time scales are combined with changes over different geographical scales, macro- and microlevel changes in fishery catches

The associate editor coordinating the review of this manuscript and approving it for publication was Eunil Park.

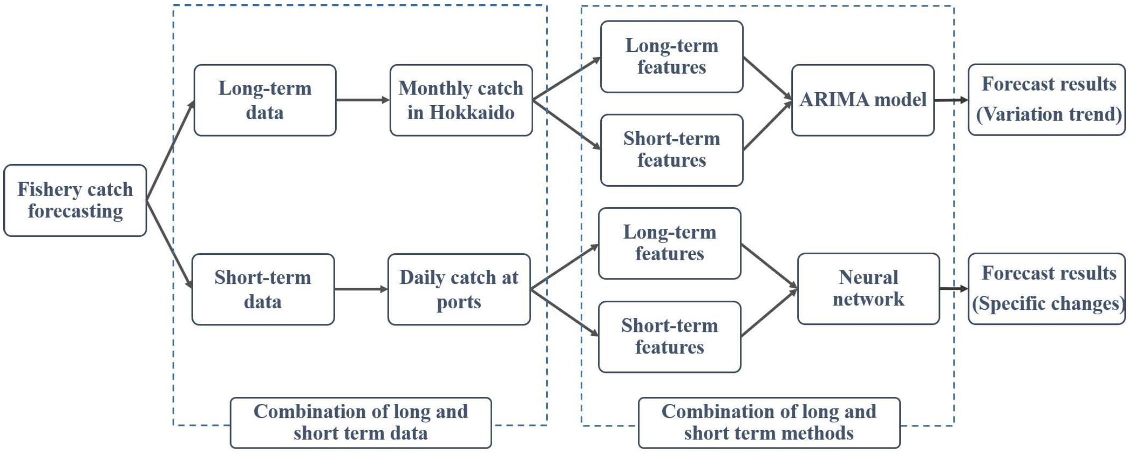

Fishery catch forecasting
Long-term data
Monthly catch in Hokkaido
Short-term data
Daily catch at ports
Combination of long and short term data
Long-term features
Short-term features
ARIMA model
Forecast results (Variation trend)
Long-term features
Short-term features
Neural network
Forecast results (Specific changes)
Combination of long and short term methods

FIGURE 1. Our process summary.

can be demonstrated more effectively. However, in existing works, long-term and short-term catch forecasting studies have often been conducted independently. Hence, analyzing and studying catch data from multiple perspectives remains challenging.

Moreover, the fishery industry is subject to problems involving limitations of available data in practical applications. In fact, in Japan, particularly in Hokkaido, numerous challenges arise in catch data collection and collation. Due to work cycles or actual work conditions, fishery practitioners responsible for catch data often do not have complete and detailed records, leading to a lack of real data and inaccuracies. This situation is particularly severe in the case of short-term catch data, for which it is not easy to achieve the desired forecasting by relying only on the analysis a single catch data type.

To solve these problems, it is necessary to integrate long-term and short-term catch data to develop a more comprehensive analytical approach. To this end, we performed data selection and considered the data management approach of the fishery industry in the Hokkaido region of Japan (Fig. 2). From the figure, it may be observed that catch data are managed in the Hokkaido region via a cascading accounting process, in which data are first collected by individual fishing vessels or ports, then aggregated to the regional fishing associations, and finally pooled from each region to construct the total catch for the Hokkaido region. This method of data management creates a difference between data recorded at high and low levels. Because the high-level data representing a large scope is composed of lower-level data representing a small scope, the statistical period for aggregating the data is longer for high-level data than for lower-level data. In other words, high-level data are generally counted only when low-level data are accumulated to a certain extent. Consequently, long-term data tend to correspond to a wide range of geographical data records. By contrast, short-term data tend to have a relatively small geographical scope.

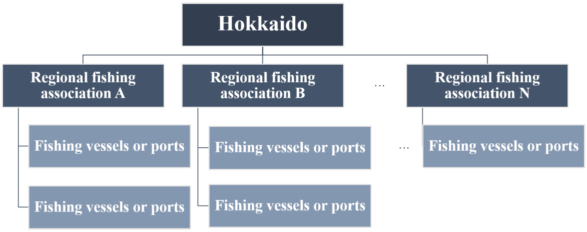


FIGURE 2. Data management approach of fishery industry in the Hokkaido.

To the best of our knowledge, the present work is the first to integrate long- and short-term data processing for fishery data. The proposed approach is implemented in the aforementioned forecasting study on the catch in Hokkaido. Monthly catch data for the entire Hokkaido province and daily catch data for the eastern ports of Hokkaido were used. The proposed approach is illustrated in Fig. 1. We also consider the combination of long- and short-term features according to the actual situation of the data; a model based on neural networks and autoregressive integrated moving average (ARIMA) [8] is proposed to handle short-term characteristics and long-term catch data, and was implemented to forecast long-term catch data. Similarly, a neural network model with a data splitter able to extract long- and short-term features from the data was used to forecast short-term catch data.

II. FISH CATCH FORECASTING

A. PROBLEM FORMULATION

Given time-series data $\mathcal{D} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \ldots\}$, each $\boldsymbol{y}_s$ is the n-dimensional real vector. Our objective is to establish the estimation of $\boldsymbol{y}_{s+T+1}$ using T-length sequence data $\{\boldsymbol{y}_{s+1}, \boldsymbol{y}_{s+2}, \ldots, \boldsymbol{y}_{s+T}\}$, where $s = 0, 1, 2, \ldots$. That is, our goal is to generate a rolling forecast of a series of future signals. Concerning the dimension of each datum, we use two cases for $n = 2$ and $n = 1$. In the case of $n = 2$, each datum is presented by $\boldsymbol{y}_s = (y_s^{\text{temp}}, y_s^{\text{catch}})$ for daily catch forecasting, In the case of $n = 1$, each datum is presented by $\boldsymbol{y}_s = y_s^{\text{catch}}$ in monthly catch forecasting. Where y_s^{temp} and y_s^{catch}) present the sea temperature data and salmon catch volume data, respectively.

B. FORECASTING OF DAILY CATCH OF PORT USING NEURAL NETWORK

In the process of applying fish catch forecasting, real time-series data often include a mixture of short- and long-term patterns. This is particularly true for port-catch data, which are highly cyclical. Long-term characteristics reflect the labor cycle, seasonal climate, and other changes, whereas short-term features reflect the effects of weather and human activities, among other influences. Accurate time-series forecasting is not possible without considering both types of cyclical patterns. In this study, we aim to solve this problem using a machine learning approach.

Hochreiter *et al.* proposed the long short-term memory (LSTM) architecture as a machine learning method based on recurrent neural networks (RNNs) [9], with certain information mining capability for long-range temporal data [10]. LSTM models are widely used in the recognition of speech, emotion, and human activity, as well as in load forecasting [11]–[14]. Behavior forecasting and the recognition of activity, speech, and emotion have also been performed by fusing convolutional neural network (CNN) [15] and LSTM [16]–[19] models. The common bases of these methods are the extraction of high-dimensional features with the short sequence feature abstraction ability of CNN models and the subsequent synthesis of high-dimensional short-sequence features for temporal forecasting, which is suitable for processing temporal data with local relevance.

On the basis of these methods, we use a specialized neural network model with a structure consisting of a convolutional layer, a data splitter, two recurrent LSTM layers, and a fully connected layer. Input layer is consisted with $n \times T$ nodes. We use T length sequence data $\mathcal{X}_s = \{\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+2}, \ldots, \boldsymbol{x}_{s+T}\}\, (s = 0, 1, 2, \ldots)$ for the case$n = 2$. The whole data set generated by the given time-series data is presented by $\{\mathcal{X}_s\}_{s>0}$. Concretely, x_s^1 and x_s^2 are the variables of the input nodes for the temperature and catch volume data, respectively.

The model uses a convolutional layer to extract high-dimensional features, a data splitter to divide the data into different time scales, and a recurrent layer to capture the complex long-term dependency patterns. The splitter divides the data into two time scales; one is input to the LSTM layers and the other is processed by skip-LSTM layers. Finally, a fully connected layer is used to integrate the outputs of the two LSTM layers and generate the final prediction results. This network architecture enables better learning of the periodicity of the input time-series data (Fig. 3). The technical details of each section are described in detail later (Section II B 1- 4).

1) CONVOLUTIONAL LAYER

The CNN model performs a convolution operation by generating convolutional check information in a receptive field of an appropriate size, which can express the original data at a higher, more abstract level. Therefore, we use the 1D CNN, which is widely used to process sequence data [20], as the convolutional layer (without a pooling layer) to extract the short-term features of the time series, where the input matrix is $\mathcal{X} \in \mathbb{R}^{n \times T}$, and the k-th filter sweeps through the input and produces $C^k = \left\{c_{s+1}^k, c_{s+2}^k, \ldots, c_{s+T}^k\right\}\, (k = 1, 2, 3, \ldots, K)$, where K is the number of filters, as shown in Eq. (1).

$$C^k = ReLU(\mathcal{X} * W_k + b_k), \tag{1}$$

where $*$ represents the convolution operation, $W_k \in \mathbb{R}^n$ is the weight vector of the convolution kernel, and b_k represents the offset. ReLU is used as the activation function, and ReLU (x) = Max (0, x). The structure of a 1D CNN is shown in the convolutional layer section of Fig. 3, with the input time-series data from the input layer, which is a multidimensional matrix $\mathcal{X} \in \mathbb{R}^{n \times T}$. It is convolved from top to bottom as shown by the arrows in the figure, with red representing one filter and green representing another, which can be followed by other filters. We make each vector C^k of length T by zero-padding for the input matrix $\mathcal{X}$ and the output matrix $\mathcal{C}$ of the convolutional layer of size $K \times T$.

2) DATA SPLITTER

LSTM models can remember historical information and understand long-term dependencies. However, longer input results in more information being contained within the system, and when the sequence is excessively long, instability and gradient disappearance occur during the training of a single LSTM. Thus, long-term interdependence cannot be captured. The proposed model mitigates this problem by using a data splitter and bypassing the loop layer, which leverages the real-world cyclical pattern.

The data splitter intervals ($p \geq 1$) longer data from the input data. In the output matrix $\mathcal{C} \in \mathbb{R}^{K \times T}$ of the convolutional layer, extracts $D_T^{\text{skip}} = \{C_{s+1}, C_{s+1+p}, \ldots, C_{s+1+\lfloor T/p \rfloor * p}\}$, where $D_T^{\text{skip}} \in \mathbb{R}^{K \times (s+1+\lfloor T/p \rfloor)}$. And the original data are used as the $D_T = \{C_{s+1}, C_{s+2}, \ldots, C_{s+T}\} \in \mathbb{R}^{K \times T}$, where $C_t = \{c_t^1, c_t^2, \ldots, c_t^K\}(t = s+1, s+2, \ldots, s+T)$. D_T is input into the LSTM layer, and the split data D_T^{skip} are then input into a new LSTM layer, S-LSTM.

3) RECURRENT LAYER

In the convolutional layer, the output enters the loop layer at the same time as the jump loop layer. The loop layer is an LSTM network that refers to the gate function, which is used to mine the time-series change rules of longer intervals in the time series. The structure is shown in Fig. 4.

D_T refers to the data value of the time t input sequence. c_t refers to a memory cell or cell state, which is the core of the network and controls the transmission of information.

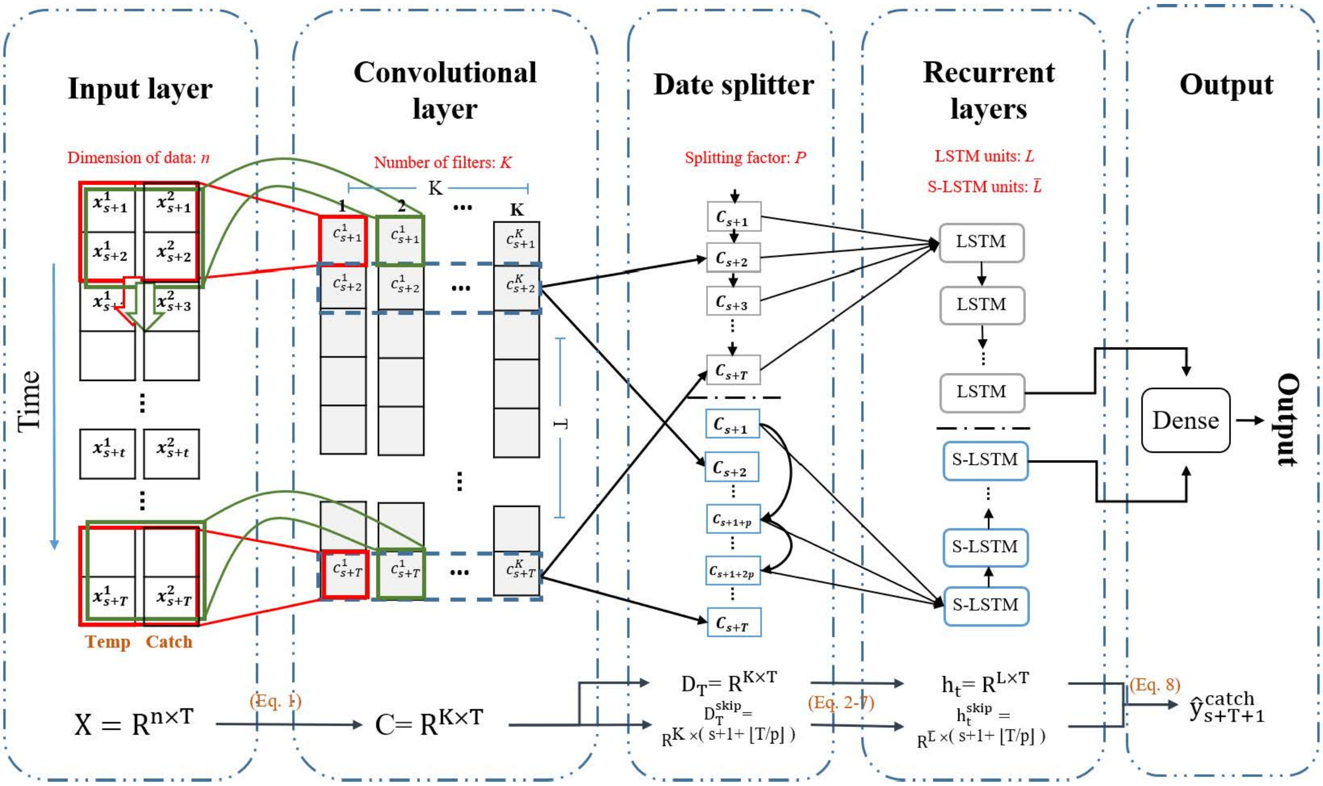


FIGURE 3. The model structure of a neural network for forecasting fish catch volume is presented. The convolutional layer, data splitter, and recurrent layer are introduced as the hidden structure between the input and output layers. The variables and parameters of the network are presented in the below. The settings of the parameters were appropriately determined for the purpose of fish catch forecasting. The convolutional layer, data splitter, recurrent layer, and fully connected layer are referred to in Section II-B-1, Section II-B-2, Section II-B-3 and Section II-B-3), respectively.

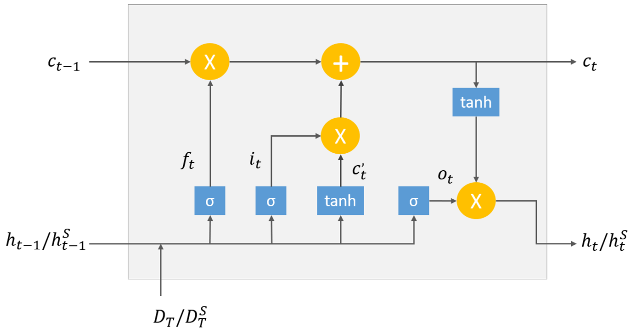


FIGURE 4. Network structure of LSTM.

i_t refers to an input gate, which determines the amount of information that the current D_T retains in c_t. f_t refers to a forget gate, which determines the amount of c_{t-1} of the cell state from the previous moment is saved to the current c_t. o refers to the output gate, which determines the amount of output h_t transmitted by c_t to the current state. h_{t-1} refers to the state of the hidden layer at time $t-1$. The corresponding formulae for the aforementioned process are provided in Eqs. (2) to (7).

$$i_t = \sigma(W^{Di}D_T + W^{h-i}h_{t-1} + b_i), \tag{2}$$

$$f_t = \sigma(W^{Df}D_T + W^{h-f}h_{t-1} + b_f), \tag{3}$$

$$o_t = \sigma(W^{Do}D_T + W^{h-o}h_{t-1} + b_o), \tag{4}$$

$$c_t^{'} = tanh(W^{Dc}D_T + W^{h-c}h_{t-1} + b_c), \tag{5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot c_t^{'}, \tag{6}$$

$$h_t = o_t \odot tanh(c_t), \tag{7}$$

where W^{Di}, W^{Df}, W^{Do}, and W^{Dc} refer to the weight matrix of the input gate, forget gate, output gate, and cell state, respectively. W^{h-i}, W^{h-f}, W^{h-o}, and W^{h-c} refer to the weight matrix of the hidden layer to the input gate, forget gate, output gate, and cell state, respectively. b_i, b_f, b_o, and b_c refer to the input gate, forget gate, output gate, and cell state offset, respectively. $\sigma(\cdot)$ refers to the *sigmoid* activation function $S(x) = 1/(1+e^{-x})$. *tanh* refers to the hyperbolic tangent activation function $tanhx = sinhx/coshx$. $\odot$ denotes the multiplying operation of the elements of a vector. The output of this layer is the hidden state of each timestamp, denoted by h_t. Here we set the dimensionality "LSTM units" of the output space to L, which means that all weight matrices W in them are in $\mathbb{R}^{L\times T}$.

The computational process of the S-LSTM layer is represented by the LSTM layer; simply replace the input data with the matrix D_T^{skip} and its output is denoted as h_t^{skip}. We set the output dimensionality of the S-LSTM layer to $\bar{L}$, so that the range of all weight matrices W is $\mathbb{R}^{\bar{L}\times(s+1+\lfloor T/p\rfloor)}$.

Finally, a fully connected layer is used to combine the output of the LSTM and S-LSTM layers, as shown in Eq. (8).

$$\hat{y}^{\text{catch}}_{s+T+1} = W^{\text{LSTM}} h_t + W^{\text{S-LSTM}} h^{\text{skip}}_t + b, \tag{8}$$

W^{LSTM} and $W^{\text{S-LSTM}}$ refer to the weight matrix of the LSTM and S-LSTM, respectively. b refers to the offset. The output $\hat{y}^{\text{catch}}_{s+T+1}$ of the fully connected layer is the forecasting result, which represents the port catch volume data.

4) LOSS FUNCTION AND OPTIMIZATION

The mean square error (MSE) [21] is the default loss function for many forecasting tasks and is calculated as shown in Eq. (9).

$$MSE = \frac{\sum_{s=1}^{m}(y^{\text{catch}}_{s+T+1} - \hat{y}^{\text{catch}}_{s+T+1})^2}{m}, \tag{9}$$

where $\hat{y}^{\text{catch}}_{s+T+1}$ is an estimation of y^{catch}_{s+T+1}. MSE denotes the MSE of the forecasting result for m days in a row. The optimization strategy adopted in this work is the same as that in the general time series forecasting model; we choose Adam [22] as the optimizer.

C. FORECASTING OF MONTHLY CATCH IN HOKKAIDO USING ARIMA

In the wide range of catch forecasting, the data volume limitation owing to the large time unit does not satisfy the most favorable conditions for neural network models, which characteristically require large amounts of data to perform training. However, these wide-range, long-time unit data, which can represent the trend changes in the catch on a macroscopic scale, are more linear and thus more suitable for processing by traditional statistical methods [23]. Therefore, we use the ARIMA model to forecast the monthly catch for the entire Hokkaido area.

Box and Jenkins introduced the ARIMA model in 1970. Also referred to as the Box—Jenkins methodology, ARIMA is composed of a set of activities for identifying, estimating, and diagnosing ARIMA models with time-series data.

The ARIMA model has three parameters, as indicated in Eqs. (10): p, d, and q. p represents the number of lags used in the forecasting model itself and is also known as the autoregressive (AR) term [24]. d refers to the time-series data that require several orders of differential differentiation to be stable and is also known as the integrated term. q represents the number of lags of the forecasting errors used in the forecasting model and is also known as the moving average (MA) term [25]. It is expressed as follows.

$$\hat{y}_t = \mu + \varphi_1 y_{t-1} + \cdots + \varphi_p y_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_p e_{t-p}, \tag{10}$$

where $\varphi_\bullet$ denotes the coefficient of AR and $\theta_\bullet$ denotes the coefficient of MA. In contrast to the daily catch forecast, we use the daily sea temperature over the month to forecast the monthly catch in the wide-area catch forecasting.

Seawater temperature has an unavoidable influence on the capture forecasting problem, but the nature of the ARIMA-based autoregression model leads to an inability to resolve other characteristic variables. Therefore, to enable the model to learn the characteristic information resulting from the water temperature data, we include an LSTM layer in ARIMA to resolve the monthly water temperature data and to enhance the forecasting accuracy, with the structure depicted in Fig. 5.

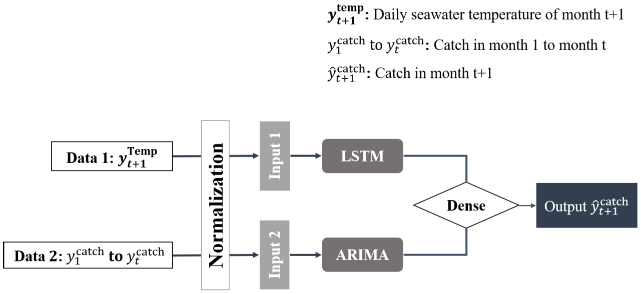

y^{temp}_{t+1}: Daily seawater temperature of month t+1
y^{catch}_1 to y^{catch}_t: Catch in month 1 to month t
$\hat{y}^{\text{catch}}_{t+1}$: Catch in month t+1
Data 1: y^{Temp}_{t+1}
Data 2: y^{catch}_1 to y^{catch}_t
Normalization
Input 1
Input 2
LSTM
ARIMA
Dense
Output $\hat{y}^{\text{catch}}_{t+1}$

FIGURE 5. Model structure of ARIMA.

First, we extract the water temperature and fish catch data as two inputs. For example, to forecast the catch data of month $t+1$, we use the catch data of the previous month 1 to month t and the daily seawater temperature data of month $t+1$.

The data normalization process has often been required for the use of the ARIMA model. Then, we substitute $(y^{\text{catch}}_t - y^{\text{catch}}_{\min})/(y^{\text{catch}}_{\max} - y^{\text{catch}}_{\min})$ with y^{catch}_t as the data normalization. Also, we substitute $(y^{\text{temp}}_t - y^{\text{temp}}_{\min})/(y^{\text{temp}}_{\max} - y^{\text{temp}}_{\min})$ with y^{temp}_t. Subsequently, we input the water temperature and fish catch data into the LSTM and ARIMA, respectively, and use two algorithms to process the two data separately. Finally, we use a fully connected layer to integrate the outputs of the two networks as the final output $\hat{y}^{\text{catch}}_{t+1}$.

III. EXPERIMENTS

A. DATA

1) DATA SOURCES

We use data from JIJI Fishery News, Fisheries Research Institute, and NEAR-GOOS. Specifically, we use the daily port catch data from JIJI Fishery News with the surface seawater temperature data from NEAR-GOOS as local-area data for the forecasting studies, whereas the monthly Hokkaido catch data from the Fisheries Research Institute and the corresponding surface water temperature data (also from NEAR-GOOS) are used as wide-area data. The details of the data are listed in Table 3.

2) TIME SPACE DISTRIBUTION OF DATA

- Short-term catch volume forecasting

We use catch data from June to November of each year from the set-net fishing periods from 2005 to 2015. The location we selected is the eastern part of Hokkaido, where the catch volume is relatively high, and selected four representative ports (as indicated in Fig. 6): Nemuro, Habomai,

TABLE 1. Details of the data used for the study.

	Data Length	Data Type	Time Unit	URL
JIJI Fishery News	2013	Set-net Fishing (Non-public data)	Daily	http://suisan.jiji.com/apps/
National Fisheries Research Institute	96	Salmon Catch	Monthly	http://salmon.fra.affrc.go.jp/zousyoku/salmon/salmon.html
NEAR-GOOS	2013/96	Surface Seawater Temperature(50m)	Daily/Monthly	https://ds.data.jma.go.jp/gmd/goos/data/database.html

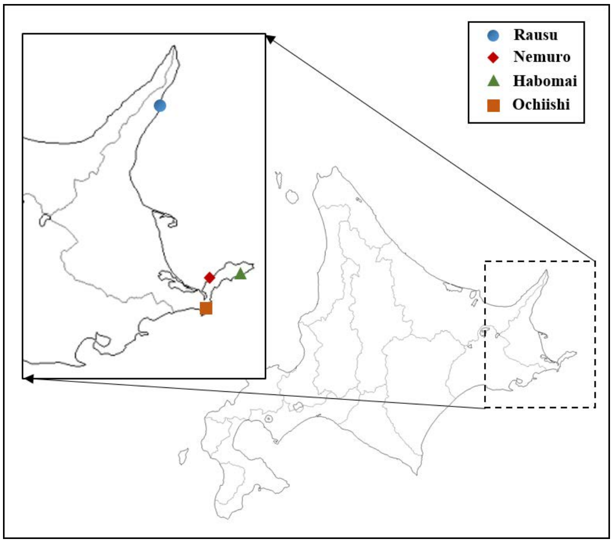


FIGURE 6. Location distribution of the four ports selected for the study.

Ochiishi, and Rausu, as well as 50 m surface water temperature data at the same locations and at the same times.

- Long-term catch volume forecasting

We select the monthly total catch data for Hokkaido from September to February of each year for the period 2000 to 2016, as well as data on the surface water temperature at 50 m for each area and time.

B. EXPERIMENTAL DETAILS

1) SHORT-TERM CATCH VOLUME FORECASTING

For this part of the experiment, as the total amount of data spanned the period 2005 to 2015, we use the port catch and water temperature data for the 2005 to 2014 period from each of the four ports as the training set for the neural network input. Moreover, we use the data for the remaining year, 2015, as the testing set to verify the experimental results. The inputs to the network are the catch and water temperature data from day 1 to day n, whereas the output datum is the catch on day $n + 1$.

2) LONG-TERM CATCH VOLUME FORECASTING

Data covering the period from September 2000 to February 2015 are used as the training set, and data from September 2015 to February 2016 are used as the testing set. The catch data from month 1 to month n are fed into the ARIMA network as inputs; the daily surface water temperature data from month $n + 1$ are fed into the neural network, whereas the catch data from month $n + 1$ are used as the output.

Notably, for the month $n + 1$ water temperature data, we divide the water temperature data into those of the Sea of Japan and the Pacific Ocean according to the actual sea area of the Hokkaido area (Fig. 7) and input these data into the neural network. Because the distribution of currents in the Sea of Japan differs from that in the Pacific Ocean, and the water temperature is affected by these currents [26], different states occur, which have different effects on the catch. Therefore,

TABLE 2. Details of parameters for the proposed method in short-term catch volume forecasting.

Parameters	Value
Dimension of data: n	2
Time steps: T	14
Number of filters in convolutional layer: K	40
Splitting factor in data splitter: P	4
LSTM units in recurrent layer: L	40
S-LSTM units in recurrent layer: $\bar{L}$	10

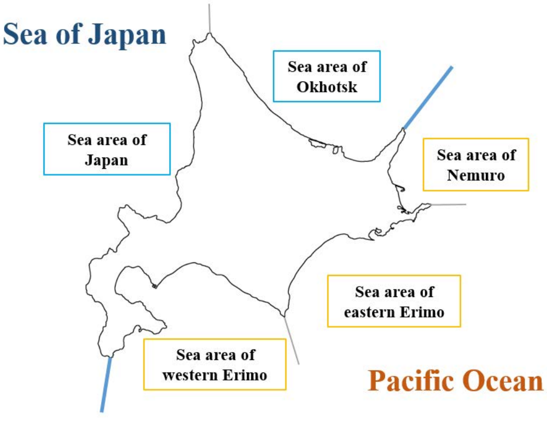


FIGURE 7. Hokkaido sea area distribution.

TABLE 3. Details of parameters for the proposed method in long-term catch volume forecasting.

Parameters	Value
Order of the autoregressive model: p	3
Degree of differencing: d	1
Order of the moving-average model: q	2
Units in LSTM layer: L	32

separating the water temperature data of these two bodies of water is more favorable for catch forecasting.

The water temperature data of the Sea of Japan were calculated by averaging the water temperature data from the sea area of Japan and those of Okhostk, whereas the water temperature data of the Pacific Ocean were calculated by averaging the water temperature data from the sea areas of Nemuro, Eastern Erimo, and Western Erimo.

3) COMPREHENSIVE ANALYSIS OF LONG-TERM AND SHORT-TERM CATCH VOLUMES

In this section, we present a comprehensive comparison of long-term and short-term data. Using techniques such as correlation analysis, we analyze the direct links between ports and between ports and catch volumes across Hokkaido. We also explore ways to link long-term data with short-term data.

C. EVALUATION CRITERIA

1) METHODS FOR COMPARISON

The methods used in our comparative evaluation are as follows.

- **XGboost** [27] is a boosting algorithm.
- **AR** refers to the autoregressive mode.
- **ARIMA** indicates the autoregressive integrated moving average model.
- **TCN** [28] is the temporal convolutional network, which combines dilated convolution and residual block.
- **CNN-LSTM** is the network model produced by combining the cell of CNN and that of LSTM.
- **LightGBM** [29] refers to a light gradient boosting machine, a distributed gradient boosting framework based on a decision tree algorithm.
- **LSTM** refers to a long short-term memory network model.
- **S-LSTM** is the proposed neural network model with a data splitter.
- **Hybrid methods** including Additive-ARIMA-LSTM, Multiplicative-ARIMA-LSTM, Additive-ETS-LSTM, and Multiplicative-ETS-LSTM four models.
- **L-ARIMA** is the proposed time-series data forecasting model based on LSTM and ARIMA.

All these methods are widely used for forecasting time-series data, covering the scope of statistics and machine learning. Among them, AR and ARIMA belong to traditional statistical methods, XGBoost and LightGBM are both gradient boosting decision trees (GBDTs) in traditional machine learning methods, and TCN belongs to deep learning. Specifically, AR and ARIMA models, as traditional statistical models, are used even more widely in forecasting efforts such as price forecasting [30], wind speed forecasting [31], and even COVID-19 situations [32]. On the other hand, XGBoost applies to both classification and regression and is used in prediction work in industries such as electricity [33] and health care [34], as well as in web text classification, malware classification [27], etc. Similarly, LightGBM has performed well in financial forecasting and cancer patient classification [35] [36]. Although TCN is a recently proposed model, it is also widely used in the fields of the weather forecasting [37], runoff forecasting [38], etc.

In addition, to validate the model's performance combining ARIMA and LSTM (L-ARIMA) is proposed in this paper. Two additive hybrid methods(Additive-ARIMA-LSTM, Additive-ETS-LSTM) and two multiplicative hybrid methods (Multiplicative-ARIMA-LSTM, Multiplicative-ETS-LSTM), which directly combine linear and non-linear models, are used for comparison.

The experiments were conducted using the TensorFlow [39] machine learning framework for the Python programming language.

2) METRICS

In this study, we used one metric, namely the root mean squared error (RMSE), to measure the magnitude of the error in the forecasting results as a quantitative criterion. The metrics are defined as follows in Eqs. (11).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m}(\hat{y}_t^{\text{catch}} - y_t^{\text{catch}})^2}{m}}. \tag{11}$$

where $\hat{y}_t^{\text{catch}}$ represents the forecast value, y_t^{catch} represents the true value, and m represents the total amount of test data. The RMSE is the square root of the ratio of the sum of the squares of the deviations of the observations from the true value to the number of observations m. It is generally used to measure the deviation of the observations from the true values.

IV. RESULTS AND DISCUSSION

A. RESULTS OF NEURAL NETWORK

Table 4 presents the forecasting errors for the catch at each of the four ports in eastern Hokkaido as predicted by the neural network. We also calculated the error percentage for comparison purposes because of the different scales of the catch volume data from port to port. This corresponds to the error after normalization of the data from the four ports and is more indicative of the model's predictive capability at each port. From the table, the S-LSTM model reduced forecasting error compared to other models commonly used for time-series data forecasting. This model greatly alleviates the unavoidable gradient disappearance and explosion problems in recurrent neural network optimization and improves forecasting accuracy by simultaneously increasing the perceptual range of the model to the data. It can be observed that the forecasting of the network model for the catch at each of the four ports fluctuated to an extent but generally remained within a specific range. The model exhibited slightly better forecasting for Nemuro compared to the other three ports, which was also reflected in the visualization results.

TABLE 4. Comparison of short-time forecasting errors. RMSE represents the actual forecast error value for each port, while RMSE[%] represents the result of converting the error value into the corresponding percentage.

Ports	Errors	XGboost	AR	ARIMA	TCN	CNN-LSTM	S-LSTM
Nemuro	RMSE	21.49	21.48	20.63	23.53	24.39	**19.43**
	RMSE[%]	3.32	3.32	3.19	3.94	3.77	**3.00**
Habomai	RMSE	24.11	22.07	26.12	30.79	23.71	**20.14**
	RMSE[%]	8.04	7.36	8.71	10.25	7.90	**6.71**
Ochiishi	RMSE	5.35	**5.30**	7.95	6.58	7.82	5.89
	RMSE[%]	4.46	**4.42**	6.63	5.49	6.52	4.91
Rausu	RMSE	90.99	82.89	87.82	84.17	98.01	**78.75**
	RMSE[%]	10.11	9.21	9.76	9.35	10.89	**8.75**

A visualization of the forecasting is shown in Fig. 8, with a comparison of the forecast and true values in 2015 for the four ports of Nemuro, Habomai, Ochiishi, and Rausu. The orange curve represents the forecast results and the blue curve represents the true changes in catch. The x-axis represents the time and the y-axis represents the catch (in tons). It can be seen that the network fit the real data well in the part where the capture was zero. In the period from October to November, when the catch volume changed dramatically, the forecasting of the capture volume data from Nemuro exhibited the best fit.

Overall, although the network showed good forecasting of the capture volume, it did not exhibit a good forecasting ability for the peak portion of the catch volume, particularly for Habomai and Rausu. Owing to the high sensitivity of the RMSE error to outliers, the RMSE [%] values for Habomai and Rausu in Table 4 were also higher than those for the other two ports. This may be attributed to insufficient feature volume data. Although the water temperature data were also included as a feature quantity in this experiment, it may be possible to capture the variation patterns of the catch quantity more effectively by incorporating richer feature quantities, such as wind speed and weather.

We also performed statistical tests of the forecast results. Concretely, t-test and likelihood-ratio test were applied for the forecast results by S-LSTM and the other methods. The p-values of the t-test results are presented in Table 5, while the p-values of the likelihood-ratio test are shown in Table 6. As a result, the p-values were greater than 0.05. Therefore, the forecast results by S-LSTM were not significantly different from the target data. Similarly, the p-values of the likelihood-ratio test are less than 0.05, and the goodness of fit to the target data of the proposed method S-LSTM is better than the other methods.

TABLE 5. *p*-values of *t*-test for S-LSTM concerning four ports.

	Nemuro	Habomai	Ochiishi	Rausu
p-value	0.233	0.405	0.334	0.657

TABLE 6. *p*-values of the likelihood-ratio test for S-LSTM against other methods (CNN-LSTM, TCN, ARIMA, AR and XGBoost).

	Nemuro	Habomai	Ochiishi	Rausu
CNN-LSTM: S-LSTM ($\times 10^{-3}$)	0.432	6.84	4.11	2.42
TCN: S-LSTM ($\times 10^{-3}$)	1.10	5.94	0.343	4.81
ARIMA: S-LSTM ($\times 10^{-3}$)	0.179	0.275	1.22	5.39
AR: S-LSTM ($\times 10^{-3}$)	0.028	1.81	46.2	8.83
XGBoost: S-LSTM($\times 10^{-3}$)	4.70	5.83	23.0	4.17

B. RESULTS OF ARIMA

The errors of the experiments are shown in Table 7. The upper two tables are comparisons of the RMSE from 2015/09 to 2016/02. The optimal estimation was not obtained by using LSTM. A plausible result was obtained by L-ARIMA. The averaged error for 10 estimations by using L-ARIMA is presented for each month in the lower table.

TABLE 7. Comparison of long-time forecasting errors. The upper two tables are comparisons of the RMSE from 2015/09 to 2016/02. The optimal estimation was not obtained by using LSTM. A plausible result was obtained by L-ARIMA. The averaged error for 10 estimations by using L-ARIMA is presented for each month in the lower table.

Methods	ARIMA	XGBoost	LightGBM	LSTM	L−ARIMA
RMSE($\times 10^7$)	4.17	4.91	5.11	non	**2.25**

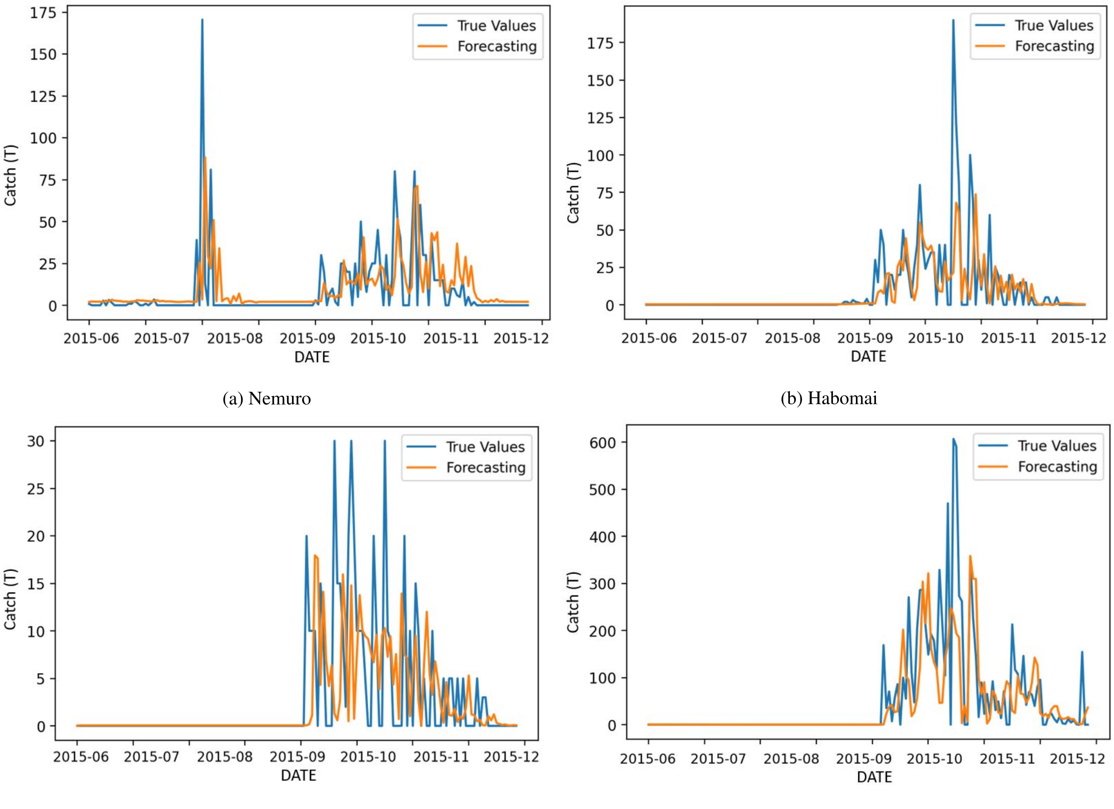
Methods	Additive ARIMA−LSTM	Multiplicative ARIMA−LSTM	Additive ETS−LSTM	Multiplicative ETS−LSTM	L−ARIMA
RMSE($\times 10^7$)	3.41	3.84	4.22	4.04	**2.25**

Year/ Month	2015/09	2015/10	2015/11	2015/12	2016/01	2016/02
RMSE($\times 10^7$) for each month by L-ARIMA	1.56	1.69	4.29	1.43	2.12	2.96

The experimental results indicate that the forecasting error is reduced after the LSTM layer, which is capable of handling the water temperature data, is added to the ARIMA model. The variation characteristics of water temperature help the network model to learn the variation patterns of the catch data. Although certain individual data are far from the real data, the forecast data still fit the real data effectively in general and could fit the trend of the real data appropriately. Moreover, compared with pure ARIMA and other forecasting methods, the L-ARIMA method achieve higher forecasting accuracy despite data volume limitations. Notably, in the case of the LSTM model used only for monthly catch forecasting, the neural network model is not able to perform learning and forecasting effectively, owing to the small amount of data.

C. ANALYSIS OF LONG-TERM AND SHORT-TERM FISH CATCH DATA

We use monthly catch data from September to early December 2015 for Hokkaido and the port catch data to analyze the correlation between the long-term and short-term catches, as shown in Fig. 10. First, although the catches of the four ports are not quantitatively the same, the trends in the timeline



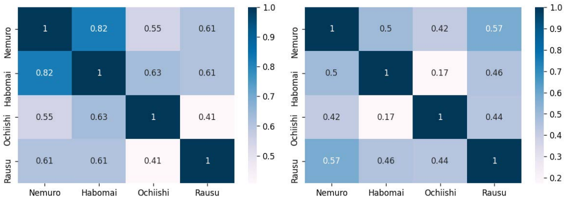
(a) Nemuro

(b) Habomai

(c) Ochiishi

(d) Rausu

FIGURE 8. **Forecasting results of daily catch per port using the proposed method.**



(a) Correlation for true data

(b) Correlation for forecast results

FIGURE 9. **Correlation of ports catch volume data, September–November 2015.**

were very similar, and the increasing and decreasing changes in the catches of each port were consistent. This is also reflected in the correlation distribution of catches between ports, as shown in Fig. 9. We analyze and compare the correlation of catch data for the four ports from September 1, 2015, to November 30, 2015. Fig. 9 (a) is the correlation plot generated from the real values of port captures, and Fig. 9 (b) is the correlation plot generated from the results obtained by our proposed network structure with two kinds of LSTM. In Fig. 9 (b), the correlations are reduced due to the errors in forecasting (especially between Habomai and Ochiishi). However, a correlation among the catches of the four ports is evident nonetheless. The variations in the catches among the four ports in the same region are influenced by geographical characteristics and are consistent to a certain extent. For this reason, the correlations can contribute to the prediction of catches in different ports in the same region. And as additional information, the reference data between a related pair of two ports in Fig. 9 might be available for the total forecasting of port catch in eastern Hokkaido.

Second, the monthly catch in Hokkaido is a cumulative value, and the magnitude of its change reflected the amount of real-time catch by combining the monthly catch of the entire island of Hokkaido and the changes in the catch data of its eastern ports. It can be observed that at the beginning of October, when the rapid increase in the monthly catch in Hokkaido approached its peak, the catch of each port also increased. Moreover, when the catch of each port decreased after the second half of October, the change in the monthly catch in Hokkaido also tended to level off. Although the port catch data are slightly inconsistent between the forecast results and the real values, it is nonetheless evident that the real-time changes in port catches are in line with the long-term trend shown by the monthly catches.

According to the aforementioned analysis, there are correlations in both the spatial and time scales between the monthly Hokkaido catch as long-term data and the port catch as short-term data. The analysis and forecasting of the two data types also play a positive role in corroborating one another and help in the analysis of the changes in the catches from different perspectives.

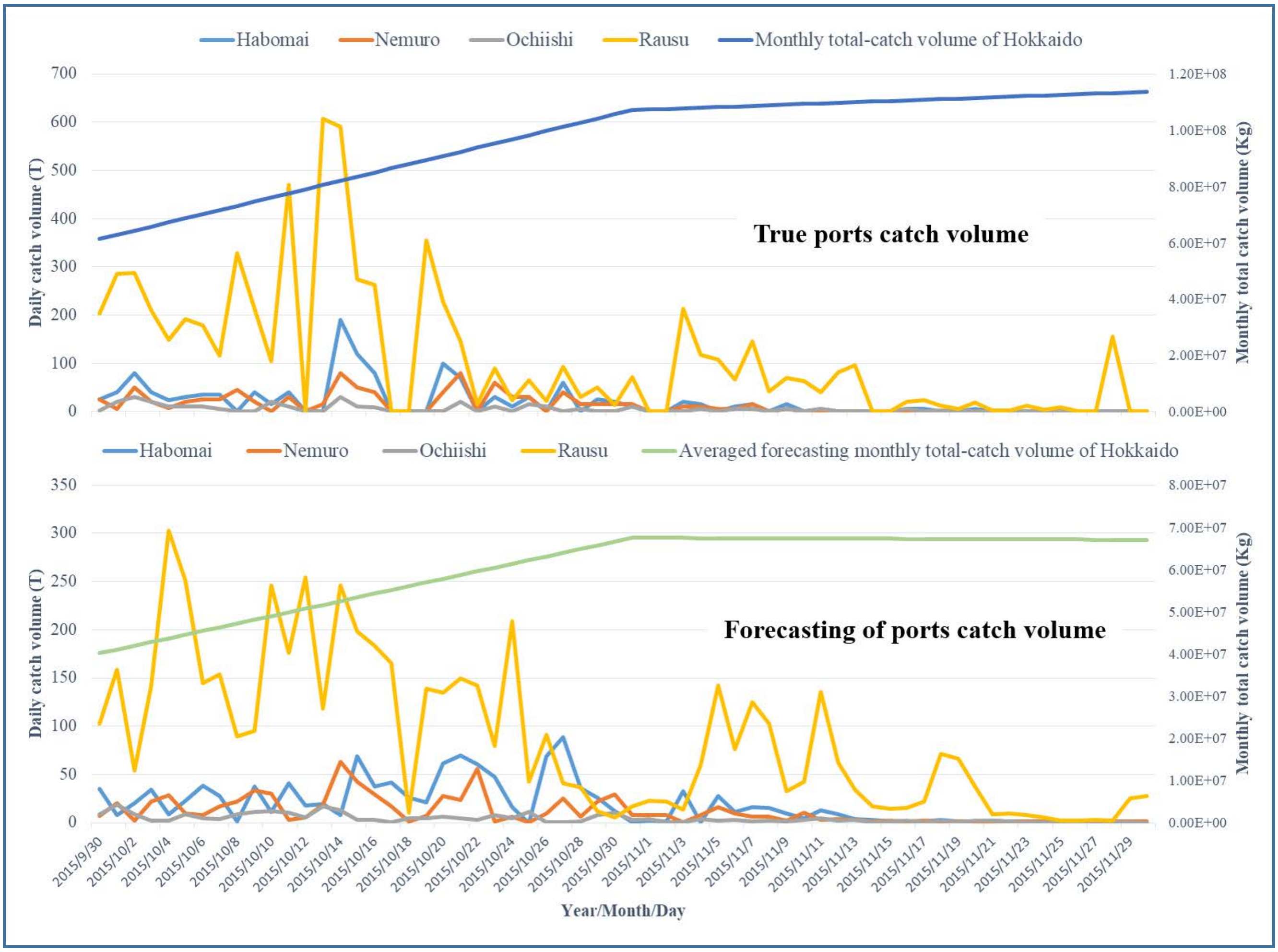


FIGURE 10. Comparison of fish catch from 2015–9 to 2015–12.

V. CONCLUSION

Two models were used to study the catch of Hokkaido. A neural network model was used to forecast the catch from the local area, short-term data, represented by the daily catch of the port, and the ARIMA model was used to study the catch forecasting from the larger regional area, long-term data, represented by the monthly catch of the entire island of Hokkaido. The results demonstrated the feasibility of the proposed approach adopting two models to forecast the data on different scales; the models were able to perform forecasting of catches with a certain accuracy. Furthermore, the forecasting results can reflect the changing patterns of the catch on different scales by forecasting and analyzing the catch from different angles, which can aid practitioners in understanding the overall data trends and reflect detailed short-term changes to guide daily fishing work.

In future work, we plan to start from the correlation of long- and short-term data and explore the possibility of using long- and short-term data to complement each other in a capture forecasting problem to achieve scaling of the data over time scales and thus improve forecasting accuracy.

REFERENCES

[1] R. Saville, K. Hatanaka, M. Sano, and M. Wada, "Application of information and communication technology and data sharing management scheme for the coastal fishery using real-time fishery information," *Ocean Coastal Manage.*, vol. 106, pp. 77–86, Mar. 2015.

[2] Fisheries Agency. *FY2019 Trends in Fisheries FY2020 Fisheries Policy Summary*. 2019. [Online]. Available: https://www.jfa.maff.go.jp/e/annualreport/attach/pdf/index-11.pdf

[3] K. Sakaguchi and N. Yamashita, "Method to forecast the catches of Japanese common squid Todarodes pacificus in the Sea of Okhotsk off Hokkaido," *Bull. Japanese Soc. Fisheries Oceanogr.*, vol. 79, no. 2, pp. 43–45, 2015.

[4] S. E. Matson, I. G. Taylor, V. V. Gertseva, and M. W. Dorn, "Novel catch projection model for a commercial groundfish catch shares fishery," *Ecol. Model.*, vol. 349, pp. 51–61, Apr. 2017.

[5] T. Komiya, "Relationship of amount of puerulus postlarvae and first-stage fry shrimp collected and japanese spiny lobster catch in chiba prefecture," *Bull. Chiba Prefectural Fisheries Res. Center*, vol. 12, pp. 27–34, 2018.

[6] J. Leathwick, J. Elith, M. Francis, T. Hastie, and P. Taylor, "Variation in demersal fish species richness in the oceans surrounding New Zealand: An analysis using boosted regression trees," *Mar. Ecol. Prog. Ser.*, vol. 321, pp. 267–281, Sep. 2006.

[7] Y. Kokaki, T. Naohiro, K. Tetsunori, H. Kazuo, H. Masayoshi, I. Akira, and T. Ogawa, "Daily fish catch forecasting for fixed shore net fishing using state space model describing probabilistic behavior of fish inside net," *IEICE Tech. Rep.*, vol. 119, no. 64, pp. 13–18, 2019.

[8] B. Kleiner, "Time series analysis: Forecasting and control," *Technometrics*, vol. 19, no. 3, pp. 343–344, Aug. 1977.

[9] A. Cleeremans, D. Servan-Schreiber, and J. L. McClelland, "Finite state automata and simple recurrent networks," *Neural Comput.*, vol. 1, no. 3, pp. 372–381, 1989.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] S. Makridakis, "Time series prediction: Forecasting the future and understanding the past," *Int. J. Forecasting*, vol. 10, no. 3, pp. 463–466, Nov. 1994.

[12] A. Baird, S. Amiriparian, M. Milling, and B. W. Schuller, "Emotion recognition in public speaking scenarios utilising an LSTM-RNN approach with attention," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Jan. 2021, pp. 397–402.

[13] A. Saeed, C. Li, M. Danish, S. Rubaiee, G. Tang, Z. Gan, and A. Ahmed, "Hybrid bidirectional LSTM model for short-term wind speed interval prediction," *IEEE Access*, vol. 8, pp. 182283–182294, 2020.

[14] C. Fan, F. Xiao, and Y. Zhao, "A short-term building cooling load prediction method using deep learning algorithms," *Appl. Energy*, vol. 195, pp. 222–233, Jun. 2017.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2012.

[16] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, Jan. 2016.

[17] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56855–56866, 2020.

[18] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 5200–5204.

[19] S. Zhang, X. Zhao, and Q. Tian, "Spontaneous speech emotion recognition using multiscale deep convolutional LSTM," *IEEE Trans. Affect. Comput.*, early access, Oct. 17, 2019, doi: 10.1109/TAFFC.2019.2947464.

[20] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, and D. J. Inman, "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks," *J. Sound Vib.*, vol. 388, pp. 154–170, Feb. 2017.

[21] E. L. Lehmann and G. Casella, *Theory of Point Estimation*. New York, NY, USA: Springer, 2006.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[23] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognit. Lett.*, vol. 42, pp. 11–24, Jun. 2014.

[24] H. Akaike, "Autoregressive model fitting for control," in *Selected Papers of Hirotugu Akaike*. New York, NY, USA: Springer, 1998, pp. 153–170.

[25] J. Durbin, "Efficient estimation of parameters in moving-average models," *Biometrika*, vol. 46, nos. 3–4, pp. 306–316, 1959.

[26] O. Leeuwenburgh and D. Stammer, "The effect of ocean currents on sea surface temperature anomalies," *J. Phys. Oceanogr.*, vol. 31, no. 8, pp. 2340–2358, Aug. 2001.

[27] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[28] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.

[29] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3146–3154.

[30] T. Kristiansen, "Forecasting Nord pool day-ahead prices with an autoregressive model," *Energy Policy*, vol. 49, pp. 328–332, Oct. 2012.

[31] P. Poggi, M. Muselli, G. Notton, C. Cristofari, and A. Louche, "Forecasting and simulating wind speed in Corsica by using an autoregressive model," *Energy Convers. Manage.*, vol. 44, no. 20, pp. 3177–3196, 2003.

[32] F. Khan, A. Saeed, and S. Ali, "Modelling and forecasting of new cases, deaths and recover cases of COVID-19 by using vector autoregressive model in Pakistan," *Chaos, Solitons Fractals*, vol. 140, Nov. 2020, Art. no. 110189.

[33] W. Wang, Y. Shi, G. Lyu, and W. Deng, "Electricity consumption prediction using XGBoost based on discrete wavelet transform," *DEStech Trans. Comput. Sci. Eng.*, Oct. 2017.

[34] M. Alim, G.-H. Ye, P. Guan, D.-S. Huang, B.-S. Zhou, and W. Wu, "Comparison of ARIMA model and XGBoost model for prediction of human brucellosis in mainland China: A time-series study," *BMJ Open*, vol. 10, no. 12, Dec. 2020, Art. no. e039676.

[35] X. Sun, M. Liu, and Z. Sima, "A novel cryptocurrency price trend forecasting model based on LightGBM," *Finance Res. Lett.*, vol. 32, Jan. 2020, Art. no. 101084.

[36] D. Wang, Y. Zhang, and Y. Zhao, "LightGBM: An effective miRNA classification method in breast cancer patients," in *Proc. Int. Conf. Comput. Biol. Bioinf. (ICCBB)*, 2017, pp. 7–11.

[37] P. Hewage, A. Behera, M. Trovati, E. Pereira, M. Ghahremani, F. Palmieri, and Y. Liu, "Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station," *Soft Comput.*, vol. 24, no. 21, pp. 16453–16482, Nov. 2020.

[38] K. Lin, S. Sheng, Y. Zhou, F. Liu, Z. Li, H. Chen, C.-Y. Xu, J. Chen, and S. Guo, "The exploration of a temporal convolutional network combined with encoder-decoder framework for runoff forecasting," *Hydrol. Res.*, vol. 51, no. 5, pp. 1136–1149, Oct. 2020.

[39] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, and S. Ghemawat, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.

YUE ZHANG was born in China, in 1993. He received the B.S. degree from the Industrial and Commercial College, Hebei University, China, in 2015, and the M.S. degree from the Muroran Institute of Technology, Japan, in 2019, where he is currently pursuing the Ph.D. degree with the Division of Information and Electronic Engineering, Graduate School. His research interests include data processing and time series analysis.

MASATO YAMAMOTO received the B.S. degree in information and electronic engineering from the Muroran Institute of Technology, Japan, in 2020, where he is currently pursuing the M.S. degree with the Division of Information and Electronic Engineering, Graduate School. His research interest includes data visualization.

GENKI SUZUKI (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronics and information engineering from Hokkaido University, Japan, in 2017, 2019, and 2021, respectively. He joined the Division of Information and Electronic Engineering, Muroran Institute of Technology, as an Assistant Professor, in 2021. His research interests include motion and sports video analysis. He is a member of the IEICE.

HIROYUKI SHIOYA received the B.S. degree in mathematics and the M.S. degree in electronics and information engineering from Hokkaido University, Japan, in 1990 and 1992, respectively, and the Ph.D. degree from the Faculty of Information System and Engineering, Graduate School, Hokkaido University, in 1995. He is currently a Professor with the Division of Information and Electronic Engineering, Muroran Institute of Technology. His research interests include mathematical science and optical informatics. He is a member of the OSA and IEICE.