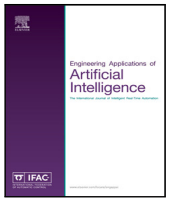
Engineering Applications of Artificial Intelligence 118 (2023) 105675



Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Assessment of the spatiotemporal prediction capabilities of machine learning algorithms on Sea Surface Temperature data: A comprehensive study



Serkan Kartal *

Geoscience & Remote Sensing Department, Faculty Of Civil Engineering and Geosciences, TU Delft, Netherlands
Department of Computer Engineering, Engineering Faculty, Cukurova University, Adana, Turkey

ARTICLE INFO

Keywords:
Machine Learning
Prediction
Time series satellite data
Sea Surface Temperature

ABSTRACT

Spatiotemporal time series prediction plays a crucial role in a wide range of applications. However, in most of the studies, spatial information was ignored and predictions were carried out either on a few points or on average values. In this study, 37 different configurations of 4 traditional ML models and 3 Neural Network (NN) based models were utilized to provide a comprehensive comparison and evaluate the spatiotemporal data prediction capabilities of the ML models. Additionally, to reveal the importance of spatial data for the time series prediction process, the best configuration of each ML model was evaluated with and without using spatial information. The utilized models were: (i) Linear Regression (LR), (ii) K-Nearest Neighbors (KNN), (iii) Decision-Trees (DT), (iv) Support Vector Machine (SVM), (v) Multi-Layer Perceptron (MLP), (vi) Long Short-Term Memory (LSTM), and (vii) Gated Recurrent Unit (GRU). The study was performed on the Sea Surface Temperature (SST) data collected by satellite radiometers via infrared measurements. The models were evaluated according to their one-month ahead spatiotemporal SST prediction performance over the southern coasts of Turkey, and the effects of spatial information on model performance were presented. Results reveal that the spatial information increased the prediction performance by approximately 25%, in terms of RMSE. Additionally, acquired results show that the LSTM model outperforms all other ML models and gives the smallest prediction errors in all metrics.

1. Introduction

Machine Learning (ML) models play a critical role in analyzing time series data and making robust predictions required for immediate and informed action (Shao et al., 2022). Recently, the increase in data collected via satellites and stations increases the importance of these algorithms in understanding climatic and environmental changes and predicting future situations (Amato et al., 2020; Koehler and Kuenzer, 2020). Sea Surface Temperature (SST) is one of these critical parameters, and it is important for understanding the interaction between the Ocean and Earth's atmosphere. Since surface water resources cover approximately 71% of the Earth's surface, SST changes have a major impact on global climate and biological systems (Bouali et al., 2017; Yao et al., 2017). Therefore, it is crucial to carry out an accurate spatiotemporal prediction of the observation data.

In recent years, many methods have been developed for time series prediction. These methods can be classified into three main categories; numerical methods, data-driven methods, and hybrid (the combinations of these two) methods (Xiao et al., 2019). Numerical methods are based on physics equations that have high computational complexity and demand massive computational efforts (Yang et al., 2018; Zhang et al., 2020). In addition, the physical environment parameters must be chosen precisely to make the numerical methods more realistic. However, these methods are not capable of making site-specific SST predictions when better exploitation of data is needed. On the other hand, data-driven methods can learn patterns directly from data and use these patterns to predict future SSTs. These methods are ranging from traditional statistical methods to the state of art Neural Network (NN) models. Traditional statistical methods include Linear Regression (LR) (Laepple Stephen Jewson, 2007), Markov models (Xue et al., 2000), Seasonal Autoregressive Moving Average (Curceac et al., 2019), and harmonic analysis (Jiakang et al., 2017). Due to its relative simplicity and well-known properties, in the literature, the LR algorithm is accepted as both a statistical method and a fundamental supervised machine-learning algorithm. Although statistical methods can predict the trend of the time series data, these methods are not effective enough for nonlinear problems, and their prediction accuracy is generally unsatisfactory (Shao et al., 2021). Thus, many researchers applied Machine Learning (ML) methods such as NNs (Ferchichi et al., 2022; Patil and Deo, 2017) and Support Vector Machines (SVM) (Aguilar-Martinez and Hsieh, 2009; Lins et al., 2013) in their studies.

* Correspondence to: Geoscience & Remote Sensing Department, Faculty of Civil Engineering and Geosciences, TU Delft, Netherlands.
E-mail addresses: s.kartal@tudelft.nl, skartal@cu.edu.tr.

https://doi.org/10.1016/j.engappai.2022.105675
Received 26 July 2022; Received in revised form 19 October 2022; Accepted 24 November 2022
Available online 9 December 2022
0952-1976/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

A close look at the literature reveals that Recurrent Neural Network (RNN) based models have started to attract more attention in time series prediction due to their higher accuracy on time series data compared to other traditional methods. These algorithms have proven their capability in learning temporal dependencies in time series datasets and predicting the next steps. Although RNN algorithms are good at analyzing time-series data, early versions of this algorithm have not been able to model long-term dependencies due to vanishing and exploding gradient problems. Therefore, a new model named Long Short-Term Memory (LSTM), which eliminates these problems thanks to its repetitive architecture and gate functions has been proposed (Hochreiter and Schmidhuber, 1997). In addition to the LSTM, its simpler and faster variant, Gated Recurrent Unit (GRU), has been developed by Cho et al. (2014). Although researchers have extensively utilized LSTM and GRU algorithms in various fields such as finance, remote sensing, and agriculture (Babu et al., 2020; Chen et al., 2018; Chhetri et al., 0000; Fan et al., 2020; Jin et al., 0000; Lin et al., 2022; Ndikumana et al., 2018; Patel et al., 2020; Persio and Di, 2017; Sezer and Ozbayoglu, 2018; Tölö, 2020), their use in the SST field is relatively new (Haghbin et al., 2021; Liu et al., 2021; Shao et al., 2021; Yu et al., 2020; Zhang et al., 2020).

Regarding the LSTM network in the SST prediction, pioneering work was conducted by Zhang et al. (2017). The study examined the performance of Support Vector Regression (SVR), MLP, and LSTM models to predict the SST in the coastal seas of China. In line with the previous work, Liu et al. (2018) explored the capabilities of LSTM, MLP, and SVR to model SST in oceans. Both studies reported that the LSTM provided the most accurate results among the considered models. Unlike the classical LSTM, Xiao et al. (2019) combined Convolutional Neural Network (CNN) and LSTM models to predict the SST based on satellite data. Further research involving the LSTM model was conducted by Xie et al. (2020), who investigated SVR, LSTM, and GRU encoder–decoder for SST prediction in the South China Sea and the Bohai Sea. So far, to our knowledge, the first and only effort to use GRU for SST prediction was conducted by Zhang et al. (2020).

Concerning the literature, most of the previous studies have only focused on temporal information, and have been conducted either on a few points or on the average values of a study area. These approaches are not capable of simultaneously considering the spatial and temporal patterns of data for site-specific studies (Yang et al., 2018). Usually, geolocation information contained in time series data is ignored. However, similar to the SST, most of the remote sensing data have their characteristics. Such that the SST of all adjacent points interacts with each other. Therefore, analyzing the temperature data by isolating the data points from each other means ignoring the SST interaction (pattern) among the points, which negatively affects the accuracy of the SST prediction. Besides, not only using spatial and temporal information as input but also predicting SST in a spatiotemporal format such as SST maps is an important factor for the applications such as sports events, fisheries, and tourism (Sarkar et al., 2020). Therefore, an accurate model capable of recognizing meaningful spatiotemporal patterns in the dataset, and similarly performing spatiotemporal prediction is greatly desired (Sun et al., 2017; Wang et al., 2017).

Although many studies in the literature indicate that they perform spatiotemporal predictions, in general, these algorithms only use these data as inputs and produce only a single output value (Alaka et al., 2018; Fan et al., 2020; Shao et al., 2022; Wang and He, 2022). In other studies where a spatiotemporal map was produced as output, mostly the LSTM model was combined with the CNN model and used as a hybrid approach (Yang et al., 2018). However, CNN models were primarily developed for tasks such as image recognition, image classification, and object detection. Thanks to their filters adjusted during the training process, these models are able to detect certain features that can be used for image classification. On the other hand, if the CNN-based models are used in remote sensing data, the number of input data that affects the temperature value on the output map will be limited by the size of the filters used. Thus, if there is a correlation between two points on the SST map and these points are farther from each other than the filter size, it will be challenging to detect this correlation with CNN filters. Therefore, since the aim of the study is to evaluate models that can be used on a large scale, CNN-based algorithms were not included in this study.

Consequently, the limitations and disadvantages of the studies in the literature can be summarized as follows; 1- Most of the current time series prediction studies suffer from using only temporal information or performing either on a few points or on the average values of the data. 2- There is a lack of studies in the literature that reveal the importance of spatial information for time series prediction. 3- There is a lack of comprehensive comparison of the performance of ML algorithms on time series prediction. Thus, three main contributions of this study to the literature can be explained as follows: 1-This study quantitatively demonstrates the advantages of spatial information for the time series prediction process . 2- A comprehensive experimental study including the new family of NN algorithms, i.e., LSTM and GRU was conducted, and the performance of the models was presented comparatively. 3- A framework for spatiotemporal prediction was proposed, which could form the basis for future studies.

2. Study area and data

The southern coasts of Turkey, located in the eastern part of the Mediterranean Sea were considered. Although many studies have been carried out in the Bohai Sea, South China Sea, East China Sea, South Atlantic Ocean etc., to the best of our knowledge, there is no previous study performed on the southern coasts of Turkey, located in the eastern part of the Mediterranean Sea. In addition, the region of interest has both temporal and spatial temperature patterns, indicating that this region is very useful for analyzing the spatiotemporal prediction performance of ML algorithms. The length of the coast is 1542 kilometers from the Syrian border in the east to Marmaris in the west. The region surrounded by the green rectangle in Fig. 1 points to the study area with a size of 205×50 pixels. Additionally, the histogram graph of the data is given in Fig. 2 along with its basic statistics information. This Figure shows the presence and frequency of temperature values in the study area for the whole date range.

In this study, a multi-sensor data acquired by Moderate-Resolution Imaging Spectroradiometer (MODIS), Advanced Very High Resolution Radiometer (AVHRR), Spinning Enhanced Visible and Infrared Imager (SEVIRI), AVHRR-3, and Advanced Along-Track Scanning Radiometer (AATSR) were used (ISAC, 2016). Dataset provided by NASA Earth Observing System Data and Information System (EOSDIS). Since the aim of the study is to evaluate the performance of ML algorithms in the spatiotemporal prediction process regardless of the observation technology, in this study, the data sources are not examined separately. All SST time series data were downloaded from the Physical Oceanography Distributed Active Archive Center (PO.DAAC). The original dataset consists of the daily gap-free maps (L4) at 0.01 deg. x 0.01 deg. horizontal resolution over the Mediterranean Sea. Since the dimensions of the original data (1025×250) belonging to the studied region are too large to be processed on the existing hardware, scaling was applied to the dataset as a preprocessing step and the size of the data was reduced to 205×50 ($W \times H$). In the resampling process, the inter-area method, which takes into account the pixel-area relationship, was applied with the help of the Python OpenCV library. In this study, daily site data from January 2008 to December 2020 were selected and reorganized by taking their monthly averages. Consequently, 156 monthly SST maps, the average of the daily temperature values measured from the beginning to the end of each month, were generated for the selected region. While the monthly average data generated for the date range 01.01.2008–31.12.2019 were used for the training&validation process, the monthly average data created for the date range 01.01.2020–31.12.2020 were used for the testing process. Moreover, 20% of the training dataset was reserved for the validation process.

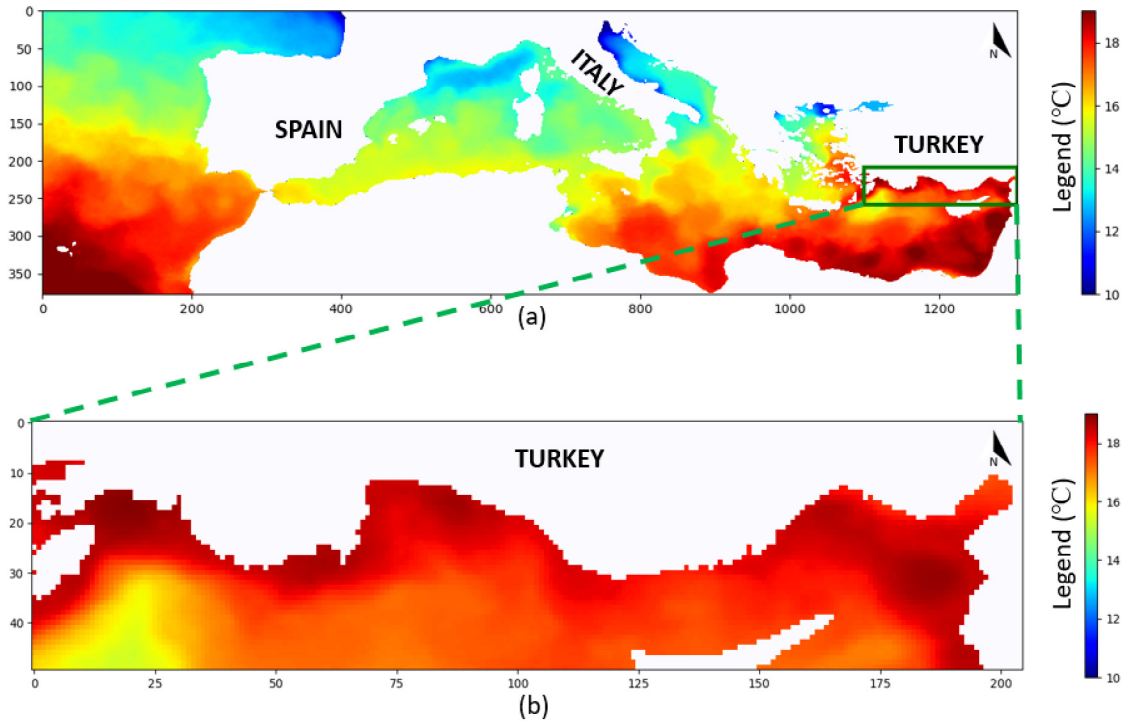


Fig. 1. The study area located in the eastern part of the Mediterranean Sea (a), covers the southern coast of Turkey (b). The image corresponds to the monthly average SST map from the National Aeronautics and Space Administration (NASA) Earth Observing System Data and Information System.

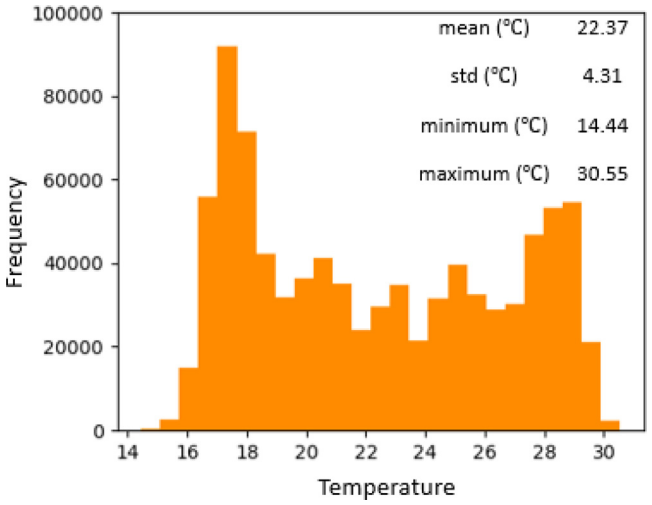


Fig. 2. The histogram graph of the temperature data along with its basic statistics information.

To consider the spatial information, each temperature value (corresponding to a pixel in Fig. 1) was handled along with its coordinate information. All the data have spatial information due to their latitude and longitude geographic locations. These spatial relative positions need to be converted into a format that ML models can easily handle. When SST data is converted into a matrix format depending on the relative position of longitude and latitude, a spatial relative relationship of the input data is revealed. Since the aim is to learn the spatial pattern according to the distance of the coordinate information to each other, in this study, row and column indexes were used instead of latitude and longitude coordinates. Similarly, month and year information were added to the data. Consequently, each data point was arranged to consist of 5 features (row, column, year, month, and SST). However, to learn temporal dependencies and to predict the SST values of the next time step, a certain length of historical observation must be added to the system. In this study, data belonging to the previous 6 months were used to predict the SST values of the one step ahead (7th month). In other words, during the training process, the model only saw the data from 01-2008 to 12-2019. To make an SST prediction for the first month of 2020 of the test data set, 6-month data for the date range 07-2019–12-2019 were given as input data (x) to the ML models, and the 01-2020 SST map (y) was predicted . The same process was repeated by shifting the date range by 1 month until the final SST map belonging to the test dataset (12-2020) was predicted. Similarly, in the last step, the data between 06-2020 and 11-2020 were given as input and the final SST map of 12-2020 was predicted.

Although the step size (length of historical observation) affects the LSTM performance, there is no ideal step size for all datasets. The step size is intuitively selected and tested based on the characteristics and size of the data set. Depending on the temporal pattern of the temperature values, there is no need to select a step size longer than 12 months. Considering that the maximum number of steps that can be taken is 12 and the minimum number of steps is 1, it is thought that 6 months of historical information will be sufficient for the prediction process. .

3. Methods

In this study, seven ML methods were utilized for SST prediction. These algorithms are Linear Regression (LR), K-Nearest Neighbors (KNN), Decision-Trees (DT), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). All the methods used are briefly described below.

3.1. Linear Regression (LR)

The LR method is used to predict a dependent variable value (y) based on the independent variables (x). The model tries to find the line that best fits the independent variable data. This line aims to minimize

the difference between the predicted and actual output values. The LR model then allows to predict how the dependent variable (y) changes depending on the independent variables (x). The advantage of the model is that it has a simple and easy to interpret mathematical formula. In this study, the LR method is considered as the basic model to compare and evaluate the results of other models.

3.2. K-Nearest Neighbors (KNN)

The KNN algorithm is one of the simple and easy-to-implement ML algorithms that can be used to solve regression and classification problems. Problems are solved by considering the k training data closest to test data, according to a given distance metric. In general, the Euclidean distance function is used for continuous values, and Hamming distance for discrete values. The Inverse Distance Weighting (IDW) method is used to define the contributions of the neighbors so that the nearer neighbors contribute more than neighbors which are further away. The only critical parameter that needs to be configured in the algorithm is k. In this study, the KNN algorithm was utilized with three different k values 5, 10, and 20.

3.3. Decision-Trees (DTs)

DT algorithm is based on the divide and conquer principle. As the name goes, it builds a tree-like classification or regression model. It divides the dataset into progressively smaller subsets, while at the same time an associated DT is progressively developed. This process is called splitting. The final tree consists of decision and leaf nodes. The decision node represents a condition on an attribute (e.g., month) and has branches that represent the outcome of the conditions. The algorithm measures the entropy of the attributes and thus decides which attribute to use in the condition expression. The critical parameter in this algorithm is deciding when to terminate the splitting process. In our study, the " minimum samples to split" parameter, which controls the number of data that should be in the sub-branch to continue the splitting process was used. While choosing a very large "minimum samples to split" value will prevent the tree from learning the data (underfitting), a very small "minimum samples to split" value will cause overfitting. In this study, the DT algorithm was utilized with three different "minimum samples to split" parameters (5, 10, and 20).

3.4. Support Vector Machine (SVM)

SVM is another famous ML algorithm that analyzes data for classification and regression. SVM divides the feature spaces into subspaces via hyperplanes and performs the prediction according to the subspace that the test data belongs. The hyperplanes are constructed during the training process by searching the maximum margin between the hyperplanes and the support vectors. Data points closest to the hyperplanes are called support vectors. All data except support vectors are ignored when calculating hyperplanes. In this study, three different kernel functions namely, polynomial, linear, and radial basis kernel functions were chosen to find hyperplanes that divide the feature spaces optimally.

3.5. Multi-Layer Perceptron (MLP)

Unlike conventional ML algorithms, the structure of the NNs model is inspired by the human brain. Therefore, in addition to the ML algorithms mentioned above, the MLP has been also utilized for comparison. The MLP is a forward-structure NN algorithm, designed to establish non-linear relationships between inputs and outputs. A typical MLP model consists of an input layer to receive data, an output layer responsible for prediction or classification, and one or more hidden layers between those two. Each hidden layer consists of neurons that receive the output of the previous hidden layer (except for the first hidden layer that receives the output of the input layer), perform simple operations, and transmit the output to the next hidden layer (except for the last hidden layer, which transmits the output to the output layer). The training process is based on the backpropagation method, which aims to iteratively minimize the error value between the actual value and the model output. Once the training process has been completed, the neural network can be used for the predictions. An example of a simple MLP architecture is given in Fig. 3.

3.6. Long Short-Term Memory (LSTM) network

So far, four different ML algorithms that are planned to be used for SST prediction have been explained. Although these ML models have been successfully applied to classification and regression problems in the literature, these approaches still have some shortcomings in analyzing large-scale time series datasets. On the contrary, the recent application of the RNN-based models (LSTM and GRU) to time series predictions has shown the strength of these models compared to other ML models (Shen, 2018). Since the use of these two models in SST prediction is quite new, in this study, more attention has been given to these two models.

In contrast with MLP, RNN-based models have internal connections to transmit the processed values to the adjacent neuron in the same layer. In other words, these models receive information from both the previous layer and the previous moment (in this study, "moment" corresponds to 1 day, 1 month, or 3 months, depending on the prediction process performed). These two input values connected to each neuron are shown in Fig. 4, where the inner structure of RNN cells is given, and in Fig. 7, where the general structure of LSTM and GRU is given. This structure makes RNNs more dynamic than standard MLPs. Thus, RNN-based models have achieved incredible success in various areas such as language modeling, translation, and time series prediction in the last few years.

LSTM is a more sophisticated kind of RNN developed to deal with the vanishing gradient problem encountered in the base model (Manaswi, 2018). In a standard RNN, each unit consists of a very simple structure such as a hyperbolic tangent (Fig. 4), whereas an LSTM unit (memory cells) consists of certain gates namely, forget gate, input gate, and output gate (Fig. 5). The forget gate decides whether past information should be deleted or stored. The input gate quantifies the importance of the new information carried by the input data. The output gate determines what output should be generated. These gates surpass the vanishing gradient problem and make LSTM suitable for learning long-term dependencies.

The entire calculation of the LSTM models can be expressed using the equations given below:

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{1}$$

$$i_t = \sigma(W_i\left[h_{t-1}, x_t\right] + b_i) \tag{2}$$

$$\mathrm{C}'_t = tanh(W_c\left[h_{t-1}, x_t\right] + b_c) \tag{3}$$

$$\mathrm{C}_t = f_t * C_{t-1} + i_t * \mathrm{C}'_t \tag{4}$$

$$o_t = \sigma(W_o\left[h_{t-1}, x_t\right] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

where f_t, i_t, and o_t are the forget, input, and output gates, respectively. W_f, W_i, W_C, and W_o are the weight parameters, h_{t-1} is the output of the previous cell, and x_t is the input of the current layer. b_f, b_i, b_C, and b_o are the corresponding biases. C_t points to the new cell state updated based on the information stored on C'$_t$ and f_t. σ and tanh represent the sigmoid function and the hyperbolic tangent activation function, respectively. The output values of the σ and tanh functions are between (0, 1) and (−1, 1), respectively.

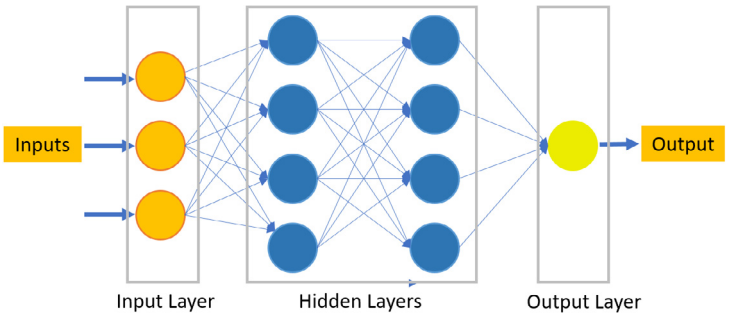


Fig. 3. An example of a simple Multi-Layer Perceptron (MLP).

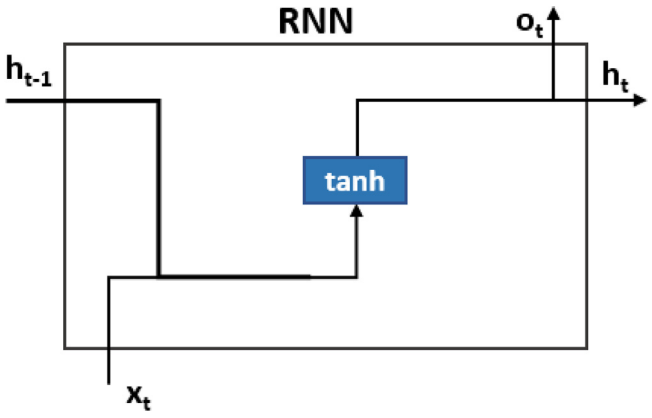


Fig. 4. The inner structure of RNN cells, where h_{t-1} is the output of the previous cell, x_t is the input of the current cell, tanh is the hyperbolic tangent activation function, o_t is the output of the current cell and input of the t_{th} cell in the next layer, h_t is the output of the current cell and input of the $(t+1)$th cell.

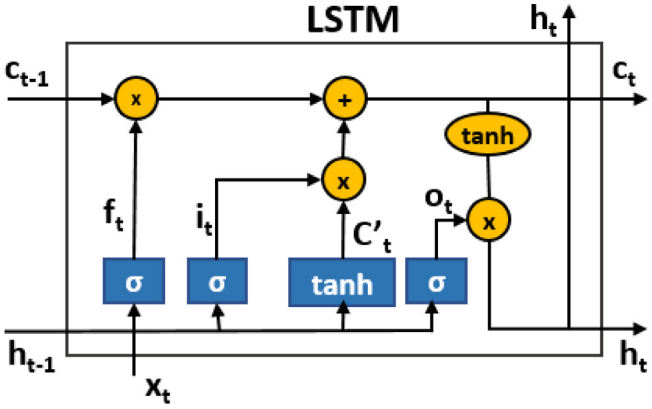


Fig. 5. The inner structure of LSTM cells.

3.7. Gated Recurrent Unit (GRU)

GRU is a relatively new variation of the LSTM and was introduced in 2014 by Cho et al. (2014). In contrast to the LSTM, the GRU model has only two gates: the update gate, which regulates the amount of past information to be transferred into the future, and the reset gate, which regulates the amount of information to be forgotten. The inner structure of a GRU cell is given in Fig. 6.

Equations in a GRU model are as follows:

$$Z_t = \sigma\left(W_z\left[h_{t-1}, x_t\right] + b_z\right) \tag{7}$$

$$r_t = \sigma\left(W_r\left[h_{t-1}, x_t\right] + b_r\right) \tag{8}$$

$$h'_t = tanh\left(W_h\left[r_t * h_{t-1}, x_t\right] + b_h\right) \tag{9}$$

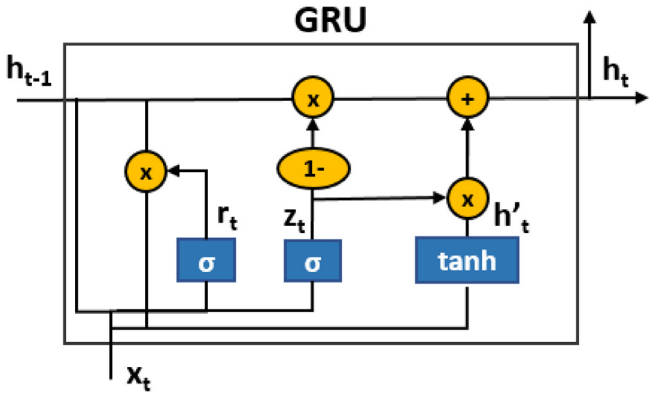


Fig. 6. The inner structure of GRU cells.

$$h_t = \left(1 - z_t\right) * h_{t-1} + z_t * h'_t \tag{10}$$

where Z_t is the update gate, and r_t is the reset gate. Similar to LSTM, x_t is the input of the current layer; h_{t-1} is the output of the previous cell; W_z, W_r, and W_h are the weight parameters; b_z, b_r, and b_h are the biases; σ and tanh are the sigmoid function and the hyperbolic tangent activation function, respectively.

3.8. Architecture of the LSTM/GRU models

In Section 3.4, the general view of the MLP architecture and the inner structures of the hidden layers and neurons were explained. On the other hand, in Sections 3.5 and 3.6, the inner structures and properties of LSTM and GRU cells were explained. However, to perform prediction operations as in MLP, a complete LSTM/GRU model having an input, an output, and several hidden layers must be constructed. Each hidden layer must consist of a limited number of LSTM/GRU cells.

Based on the LSTM and GRU architectures, NN models having a different number of hidden layers and a different number of neurons were built. The general architecture of the LSTM/GRU models utilized for SST prediction is given in Fig. 7.

4. Experimental setup

Experiments have been conducted with seven ML models. The overall workflow of an experiment is given in Fig. 8. The workflow steps are as follows: Data are rescaled according to the given scale factor. Then the whole dataset is split into two subsets: training&validation (80%), and testing (20%) sets. Training&validation set is used to train the ML models. During the training process, the termination criterion is checked after each epoch. Accordingly, if the maximum number of epochs is reached, or the validation error not decreases for a certain period of time, the training process is terminated. In the last step, the prediction of the test dataset is performed.

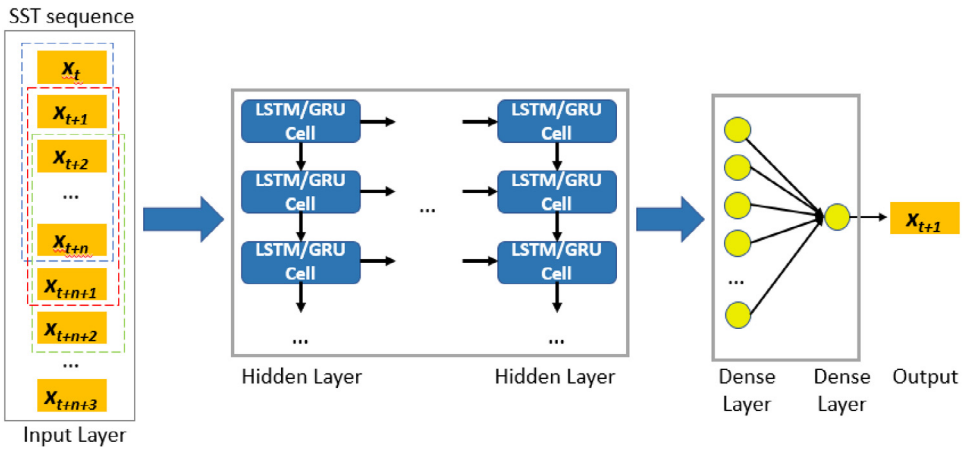


Fig. 7. The architecture of the utilized LSTM/GRU models for SST prediction.

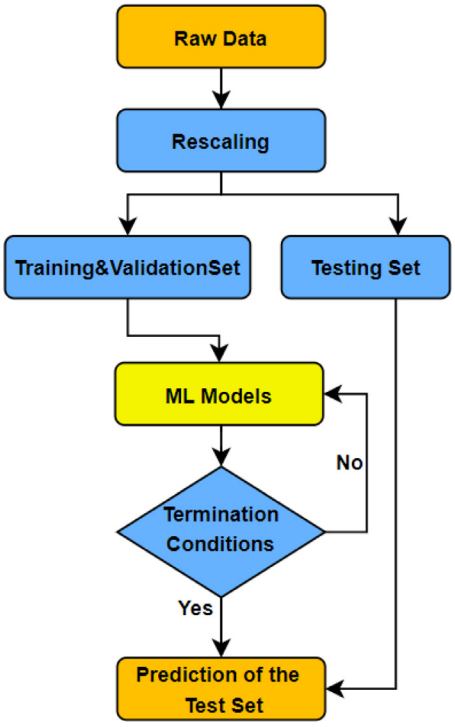


Fig. 8. General workflow of the proposed methodology.

These models need some critical parameters to be optimized to achieve accurate SST predictions. The selection of optimal parameters directly affects the performance of the models and promotes higher accuracy. However, since the parameters used in ML algorithms vary depending on the size and characteristics of the data set, these parameters are usually decided based on the user's experience, intuition, and literature knowledge. Therefore, different parameters were applied and the models were trained and evaluated based on them. However, since it is not possible to test all configurations of the models by changing each parameter separately, only the most critical parameters discussed in the literature have been modified. The remaining parameters were used with their standard configurations. Thus, the most critical parameters of traditional ML models were considered. The KNN algorithm was implemented with three different K values (3, 5 and 10), which is the most critical parameter and corresponds to the number of neighbors to be considered. If the value of k is chosen too small, means that insufficient neighbor data will be used for temperature prediction, and in this case, the model will not have enough generalization ability. On the other hand, if the k value is chosen too big, for example the same size as the data set, then the result will be the average of all the data and the entire temperature map will have the same value. The DT algorithm was evaluated with three different "minimum samples to split" values, 5, 20, and 50, respectively. A similar situation in the KNN algorithm is also valid for the DT algorithm. While a too small value of "minimum-samples split" may cause overfitting, a too large "minimum -samples split" value may cause underfitting. Furthermore, the SVM was implemented with three different kernel functions (linear, polynomial, and radial basis kernel functions). The kernel function is a mathematical function used to convert input data into the required data format. Detailed information about the kernel functions can be found in the reference (Tharwat, 2019).

Since MLP, LSTM and GRU are NN-based models, the configurations of these models are different from the traditional models. For these models, the maximum number of epochs was set to 100. In addition, an early stopping condition was determined to achieve a better generalized NN and to prevent overfitting. Therefore, a validation loss was calculated by predicting unused data from the validation dataset at each epoch. If there was no improvement in validation loss for 10 consecutive epochs, the learning rate was multiplied by 0.1 to achieve a more precise learning process. Additionally, if there was no improvement in validation loss during the patience periods (30 epochs), the training process was terminated. Mean Square Error (MSE) was used as a loss function. Considering the performance comparisons of the optimizers in the literature,in this study, Adaptive Moment (ADAM) optimizer was used to update the weights. Detailed information about the optimizers can be found in the reference (Soydaner, 2020).

Each NN model was tested with 9 different configurations that differed in (i) the number of hidden layers (1, 3, and 5) and (ii) the number of neurons (5, 10, and 20). All NN models were terminated with a single neuron output layer responsible for the output. The number of layers and the number of neutrons are parameters that determine the complexity and learning capacity of the model. While using a simple model may cause an underfitting problem, using a very complex model may cause an overfitting problem. All these algorithms were implemented using Python (Cobos et al., 2022). While KNN, SVM, and DT models were implemented with the Scikit-Learn library, the MLP, LSTM, and GRU models were implemented with Keras Library on top of Google TensorFlow (https://scikit-learn.org/stable/, accessed 07 October 2022; https://keras.io/, accessed 07 October 2022; https://www.tensorflow.org/, accessed 07 October 2022).

5. Performance evaluation metrics

Two well-known evaluation metrics, Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were used to evaluate the performance

Table 1
Average RMSE and MAE of 37 configurations of 7 ML models.

	Layer count	Neuron count	RMSE (°C)	MAE (°C)
LSTM	1	5	0.6	0.53
	2	5	**0.53**	**0.46**
	3	5	0.59	0.51
	1	10	0.65	0.57
	2	10	0.69	0.63
	3	10	0.75	0.68
	1	20	0.7	0.62
	2	20	0.82	0.75
	3	20	0.72	0.64
GRU	1	5	0.74	0.66
	2	5	0.58	0.51
	3	5	0.59	0.52
	1	10	0.74	0.66
	2	10	0.61	0.54
	3	10	0.83	0.75
	1	20	0.73	0.65
	2	20	0.74	0.67
	3	20	0.64	0.55
MLP	1	5	0.59	0.52
	2	5	0.63	0.56
	3	5	0.69	0.62
	1	10	0.57	0.55
	2	10	0.75	0.68
	3	10	0.88	0.81
	1	20	0.62	0.55
	2	20	0.7	0.62
	3	20	1.22	1.13
KNN	Neighbor count			
	5		0.73	0.64
	10		0.72	0.64
	20		0.72	0.64
DT	Min. sample count			
	5		0.84	0.74
	10		0.86	0.75
	20		0.86	0.75
SVM	Kernel function			
	Linear		0.81	0.72
	Polynomial		1.08	0.98
	Radial Basis		0.95	0.86
LR			0.78	0.71

of the models. Equations of the metrics are given below.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{11}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{12}$$

where y_i and $\hat{y}_i$, are the ith actual value and the ith predicted value among n samples. The smallest values of RMSE and MAE indicate the highest prediction accuracy.

6. Experiment results and discussion

Table 1 gives the evaluation scores of the one-month ahead SST predictions based on 7 different ML models including LR, KNN, DT, SVM, MLP, LSTM, and GRU. To make a better evaluation, the error values of the models with different configurations are also given. Except the basic LR, while each NN model was analyzed with 9 different configurations, each traditional ML model was analyzed with 3 different configurations. Consequently, a total of 37 different configurations of 7 different models were tested. Results reveal the performance of the models on the test dataset. Two statistical evaluation metrics (RMSE and MAE) were used to quantitatively evaluate the prediction performance of the models. These metrics demonstrate the 12-month average prediction errors for 2020. The evaluation was performed between predicted SST and observed SST. The best results for each evaluation metric are pointed out in bold.

According to the results given in Table 1, it is evident that all models can predict temperature values with less than 1 °C differences (MAE). This indicates that the achieved prediction results are highly correlated with the actual monthly SST data. Nonetheless, none of the traditional ML models (KNN, DT, and SVM) were able to perform predictions with less than 0.64 MAE error. Among these models, the lowest accuracy scores were produced by DT. Compared to DT, the SVM model achieved slightly better results. KNN model outperforms both DT and SVM models by a considerable margin (0.1 and 0.08) in terms of MAE scores. Besides, even if different configurations of the models were employed, these configurations did not have a significant effect on the test results. For example, MLP and LSTM methods were tested with more layers and number of neurons, but these tests were not included in the study due to higher RMSE error values. All three different configurations of the KNN model produced almost the same prediction errors. The primary reason for this problem is that time-series data has a time-dependent pattern in itself, and traditional methods have limited ability to learn this pattern. Therefore, although successful results have been achieved with traditional models, more capable algorithms such as NN are needed to achieve even better results in such datasets that have a time-dependent pattern.

When the results are examined in detail, it is clearly seen that all the NN models significantly improve the prediction accuracy compared to

Table 2
Std of monthly average temperature values, and one-month-ahead SST prediction performances of LSTM, GRU, and MLP models in terms of RMSE and MAE.

Month		LSTM		GRU		MLP	
	STD	RMSE	MAE	RMSE	MAE	RMSE	MAE
1	0.80	0.43	0.35	0.49	0.41	0.45	0.39
2	0.41	0.36	0.29	0.29	0.25	0.45	0.38
3	0.28	0.22	0.19	0.28	0.23	0.37	0.32
4	0.42	0.23	0.17	0.25	0.22	0.38	0.31
5	0.50	0.30	0.24	0.27	0.21	0.28	0.21
6	0.76	1.68	1.65	1.91	1.88	1.68	1.64
7	0.96	0.78	0.69	0.96	0.87	0.95	0.87
8	0.90	0.40	0.25	0.66	0.59	0.39	0.25
9	0.92	0.65	0.62	0.48	0.40	0.69	0.64
10	1.05	0.41	0.32	0.42	0.33	0.44	0.36
11	0.94	0.46	0.35	0.42	0.33	0.55	0.43
12	0.67	0.44	0.36	0.49	0.38	0.51	0.40

traditional ML models. The performance of the basic LR model is somewhere between NN algorithms and other ML algorithms. Compared to the best KNN results, the MAE scores of the LSTM, GRU, and MLP models decreased approximately 28%, 20%, and 19%, respectively. These results are not surprising given the strengths of the NN architectures, especially the LSTM and GRU architectures on time series data sets. Besides, these results reveal the great potential in the GRU, which has not been adequately tested in a similar study before. Although the difference was not huge, the GRU had a slightly better performance than the MLP model. On the other hand, the LSTM model still had the best performance. Overall, considering both RMSE and MAE metrics, the prediction performances of all models ranked as LSTM > GRU > MLP > KNN > LR > SVM > DT.

To further assess the prediction capability of the NN models and examine the effect of the hidden layers structure, nine different configurations of each model were analyzed. Since the structure of the hidden layers has a huge impact on the model output, both the number of hidden layers and the number of neurons in each hidden layer must be carefully decided. Employing a hidden layer structure that is too simple or too complex may cause underfitting or overfitting issues. In addition, the existing dataset should be large enough to train the designed model. Trying to train a too complex model with a small dataset may also cause an underfitting problem. Therefore, this balance should be taken into account when deciding on the hidden layer structure. According to the results, the best predictions for the LSTM and GRU models were achieved with two hidden layer structures containing five neurons. In MLP, on the other hand, the best predictions were achieved with a single hidden layer structure containing five neurons. The first reason is that the existing dataset is not large enough to train complex NN models, or that even relatively simple NN models have the capacity to learn the existing SST pattern. The second problem, which is frequently mentioned in the literature, is the vanishing gradient problem. As the number of layers increases in the NN, the value of the product of derivative decreases, and the partial derivative vanishes (Manaswi, 2018).

In order to show the one-month-ahead prediction capability of the LSTM model, the monthly average MAE scores of the models for the test dataset (1.1.2020–1.1.2021) are given in Table 2. In order to see whether there is a correlation between error values and temperature variability, the standard deviations (std) of the monthly average temperature values are also given in the table. It can be seen that there is no correlation between std and error values. Besides, the results given in this table indicate that the predictions of the SST variation, except for the sixth month, are satisfactory and agree with the observed SST. For eleven months the MAE error of the LSTM was between 0.17 and 0.69, and a relatively higher prediction error occurred in only the 6th month. To further prove the effectiveness of making one-month ahead spatiotemporal SST prediction by the LSTM and GRU, the maps of ground truth SST and predicted SSTs are presented in Fig. 9. This way, models can be verified not only in temporal but also in spatial dimensions. Each row in the figure corresponds to a month in the test data set. As seen from the figures, LSTM and GRU models can make effective predictions by learning various temperature patterns in spatial dimensions. These visual results indicate that the SST maps predicted with these models perfectly match the original monthly SST maps, and the two time series models are trustworthy.

In the literature, researchers have mainly focused either on a few points or on the average SST values of the study area. Unlike spatiotemporal SST predictions, point-based or average SST predictions do not take into account spatial information or the relationship between spatial SST data. Thus, these algorithms are easier to implement, but are not capable of providing sufficient information for studies where spatial temperature pattern is required. In those studies, daily, weekly, monthly and seasonal predictions have been conducted. In a recent study, Liu et al. (2021) predicted 1 day, 3 days, and 7 days ahead SST in the East China Sea with RMSE of 0.025, 0.061, 0.32, and in the South China Sea with RMSE of 0.013,0.041, 0.238. They utilized the B-spline interpolation and refining spatiotemporal attention mechanism along with the LSTM. In the same area (China Sea), Yu et al. (2020), predicted the SST values of 3 selected points for 3 days, 5 days, 1 week, 2 weeks, and 1 month ahead, with RMSE of 0.450, 0.457, 0.55, 0.663, 0.672, respectively. These findings indicate that, as the prediction interval increases, the error rate increases. Zhang et al. (2020) achieved the highest accuracy with an RMSE of 0.65 for 1 month ahead SST prediction in the Bohai Sea, on the six selected locations. In another study, Yang et al. (2018) combined CNN and LSTM to make a spatio-temporal prediction on a National Oceanic and Atmospheric Administration High-Resolution SST dataset covering the Bohai Sea and a dataset covering the Chinese Coastal waters. They predicted the 1, 7, and 30 days ahead SST for the Bohai Sea with the RMSE of 0.146, 0.277, and 0.726, respectively. In the same study, SST values of the coastal area of the China Sea for 1, 7, and 30 days were predicted with the RMSE of 0.40, 0.63, and 1.07, respectively. Since of the open seawater body is more stable than in the coastal area, the accuracy of the studies carried out in these areas is also higher than in the coastal areas. On the other hand, there is no acceptable exact error rate for SST in the literature. Therefore, the prediction error values obtained in these similar recent studies were taken into account in the evaluation process.

Considering these studies in the literature, some points can be highlighted. Firstly, it is clearly seen that as the predicted time interval increases, the amount of error increases, that is, 1-month ahead prediction performed in the same region with the same method produces higher error than 1-day ahead prediction. Secondly, RMSE was used as the common measurement metric in all studies. Thirdly, the minimum and maximum error values obtained for the all predictions in these studies ranged from 0.146 (°C) to 1.07 (°C). Therefore, methods that perform prediction error of less than 1 (°C) RMSE can be considered as valuable for the literature. Based on these results, in our study, RMSE and MAE metrics were used. Although our study was conducted in the

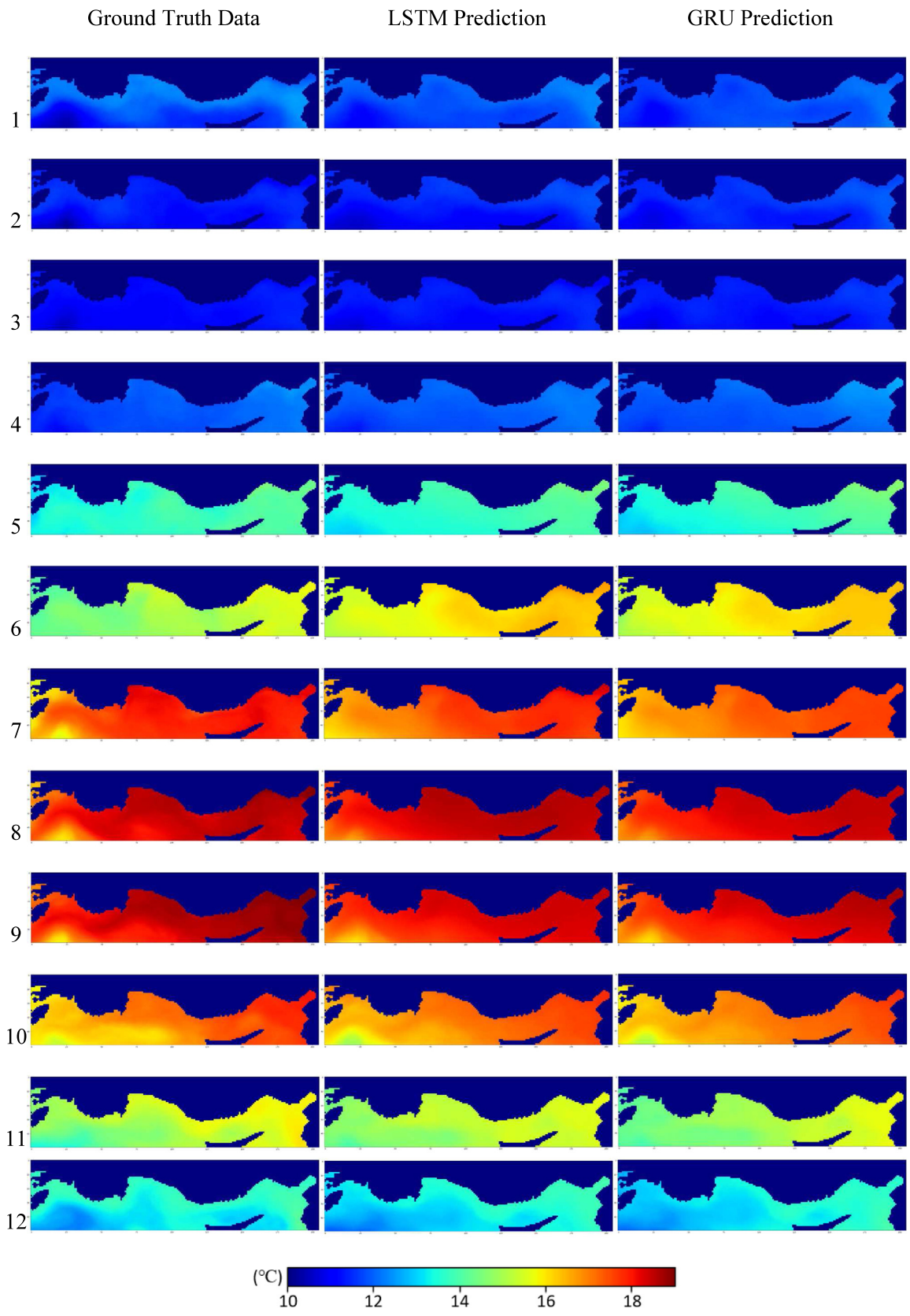


Fig. 9. Comparison of original monthly SST and predicted SST images for 12 months.

Table 3
SST prediction performances of the ML models for different configuration of the dataset.

	Layer count	Neuron count	1-Day	**1-Month**	3-Months	1-Month without spatial info.
LSTM	2	5	0.335	0.53	0.649	0.713
GRU	2	5	0.38	0.58	0.685	0.748
MLP	1	5	0.395	0.59	0.675	0.764
KNN	Neighbor count					
	10		0.4936	0.72	0.897	0.935
DT	Min. sample count					
	5		0.516	0.84	0.887	0.951
SVM	Kernel function					
	Linear		0.5265	0.81	0.858	0.853
LR			0.532	0.78	0.827	0.855

coastal area, it is seen that a successful prediction process was carried out compared to the studies in the literature.

Furthermore, to present the effect of time interval on the result as given in the literature, and to reveal how spatial information affects the prediction performance, the study was extended by testing on different configurations of the dataset. Firstly, the time interval of the models was changed and 1-day and 3-months temperature predictions were carried out. However, considering the number of tests performed and the time and hardware constraints, the related tests were carried out only on the configurations that gave the best results for 1-month data. Test results are given in Table 3 in the RMSE metric. When the results are examined, it is clearly seen that as the time interval decreases, the amount of prediction error decreases. The best RMSE error value obtained with LSTM decreased by about 0.2 in the 1-day prediction, and increased by about 0.12 in the 3-month prediction. Secondly, 1-month estimation was performed again by removing spatial information from the monthly data, that is, using only temporal information. The results show a significant increase in the amount of error. The best RMSE error value obtained with LSTM increased from 0.53 to 0.71 (~40%). Similar error increase can be seen in the results of all ML models (in Table 3). These results numerically reveal that the spatio-temporal data provides more accurate prediction than using only temporal data.

7. Conclusion

In this study, one month ahead spatiotemporal SST prediction capability of 4 traditional ML models (SVM, KNN, DT) and 3 NN models (MLP, LSTM, and GRU) were examined and the comparative analysis report was provided. Models were evaluated on the SST time series dataset covering the southern coasts of Turkey, located in the eastern part of the Mediterranean Sea. The main conclusions are as follows:

1. While NN-based models (LSTM, GRU, and MLP) outperform traditional ML models, time series models (LSTM and GRU) outperform traditional MLP and, in these two time series models, the complex one (LSTM) outperforms the simpler one (GRU). Consequently, the prediction performances of the models ranked as LSTM > GRU > MLP > KNN > LR > SVM > DT.
2. In the one-month ahead SST prediction, RMSE, and MAE of the LSTM and GRU are mostly concentrated in the range of 0–0.5 °C. This result indicates that the RNN-based models are universal and reliable models for spatiotemporal SST prediction.
3. It has been observed that the use of spatiotemporal information in the prediction process significantly reduces the error values compared to using only temporal information.
4. In the SST prediction, it is observed that as the time interval decreases, the amount of prediction error decreases.

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.

Data availability

Data will be made available on request.

References

Aguilar-Martinez, S., Hsieh, W.W., 2009. Forecasts of tropical Pacific Sea surface temperatures by neural networks and support vector regression. Int. J. Oceanogr. 13, http://dx.doi.org/10.1155/2009/167239.

Alaka, H.A., Oyedele, L.O., Owolabi, H.A., Bilal, M., Ajayi, S.O., Akinade, O.O., 2018. A framework for big data analytics approach to failure prediction of construction firms. Appl. Comput. Inform. http://dx.doi.org/10.1016/j.aci.2018.04.003.

Amato, F., Guignard, F., Robert, S., Kanevski, M., 2020. A novel framework for spatio-temporal prediction of environmental data using deep learning. Sci. Rep. 10, 1–11. http://dx.doi.org/10.1038/s41598-020-79148-7.

Babu, R.G., Maheswari, K.U., Zarro, C., Parameshachari, B.D., Ullo, S.L., 2020. Land-use and land-cover classification using a human group-based particle swarm optimization algorithm with an LSTM classifier on hybrid pre-processing remote-sensing images. Remote Sens. 12, 1–28. http://dx.doi.org/10.3390/rs12244135.

Bouali, M., Sato, O.T., Polito, P.S., 2017. Temporal trends in sea surface temperature gradients in the South Atlantic Ocean. Remote Sens. Environ. 194, 100–114. http://dx.doi.org/10.1016/j.rse.2017.03.008.

Chen, S., Li, B., Cao, J., Mao, B., 2018. Research on agricultural environment prediction based on deep learning. Procedia Comput. Sci. 139, 33–40. http://dx.doi.org/10.1016/j.procs.2018.10.214.

Chhetri, M., Kumar, S., Roy, P.P., Kim, B.-G., 0000. remote sensing Deep BLSTM-GRU Model for Monthly Rainfall Prediction: A Case Study of Simtokha, Bhutan, http://dx.doi.org/10.3390/rs12193174.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: EMNLP 2014-2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. pp. 1724–1734. http://dx.doi.org/10.3115/v1/d14-1179.

Cobos, M., Otiñar, P., Magaña, P., Lira-Loarca, A., Baquerizo, A., 2022. MarineTools.temporal: A Python package to simulate Earth and environmental time series. Environ. Model. Softw. 150, 105359. http://dx.doi.org/10.1016/j.envsoft.2022.105359.

Curceac, S., Ternynck, C., Ouarda, T.B.M.J., Chebana, F., Niang, S.D., 2019. Short-term air temperature forecasting using Nonparametric Functional Data Analysis and SARMA models. Environ. Model. Softw. 111, 394–408. http://dx.doi.org/10.1016/j.envsoft.2018.09.017.

Fan, Y., Xu, K., Wu, H., Zheng, Y., Tao, B., 2020. Spatiotemporal modeling for nonlinear distributed thermal processes based on KL decomposition, MLP and LSTM network. IEEE Access 8, 25111–25121. http://dx.doi.org/10.1109/ACCESS.2020.2970836.

Ferchichi, A., Abbes, A., Ben Barra, V., Farah, I.R., 2022. Forecasting vegetation indices from spatio-temporal remotely sensed data using deep learning-based approaches: A systematic literature review. Ecol. Inform. 68, 101552. http://dx.doi.org/10.1016/j.ecoinf.2022.101552.

Haghbin, M., Sharafati, A., Motta, D., Al-Ansari, N., Noghani, M.H.M., 2021. Applications of soft computing models for predicting sea surface temperature: a comprehensive review and assessment. Prog. Earth Planet. Sci. 8, http://dx.doi.org/10.1186/s40645-020-00400-9.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

ISAC, 2016. Mediterranean Sea ultra high resolution SST L4 analysis 0.01 deg resolution. http://dx.doi.org/10.5067/GHOUH-4GM20.

Jiakang, L., Ying, Z., Honglin, L., Oijie, L., 2017. SST forecast based on BP neural network and improved EMD algorithm. Clim. Environ. Res. 22, 587–600. http://dx.doi.org/10.3878/j.issn.1006-9585.2017.16180.

Jin, X.-B., Yang, N.-X., Wang, X.-Y., Bai, Y.-T., Su, T.-L., Kong, J.-L., 0000. Hybrid Deep Learning Predictor for Smart Agriculture Sensing Based on Empirical Mode Decomposition and Gated Recurrent Unit Group Model. http://dx.doi.org/10.3390/s20051334.

Koehler, J., Kuenzer, C., 2020. Forecasting spatio-temporal dynamics on the land surface using earth observation data—a review. Remote Sens. 12, 1–34. http://dx.doi.org/10.3390/rs12213513.

Laepple Stephen Jewson, T., 2007. Five year ahead prediction of Sea Surface Temperature in the Tropical Atlantic: a comparison between IPCC climate models and simple statistical methods.

Lin, H., Gharehbaghi, A., Zhang, Q., Band, S.S., Pai, H.T., Chau, K.W., Mosavi, A., 2022. Time series-based groundwater level forecasting using gated recurrent unit deep neural networks. Eng. Appl. Comput. Fluid Mech. 16, 1655–1672. http://dx.doi.org/10.1080/19942060.2022.2104928.

Lins, I.D., Araujo, M., Moura, M.D.C., Silva, M.A., Droguett, E.L., 2013. Prediction of sea surface temperature in the tropical Atlantic by support vector machines. Comput. Statist. Data Anal. 61, 187–198. http://dx.doi.org/10.1016/j.csda.2012.12.003.

Liu, J., Jin, B., Yang, J., Xu, L., 2021. Sea surface temperature prediction using a cubic B-spline interpolation and spatiotemporal attention mechanism. Remote Sens. Lett. 12, 478–487. http://dx.doi.org/10.1080/2150704X.2021.1897182.

Liu, J., Zhang, T., Han, G., Gou, Y., 2018. TD-LSTM: Temporal dependence-based LSTM networks for marine temperature prediction. Sensors (Switzerland) 18, 1–13. http://dx.doi.org/10.3390/s18113797.

Manaswi, N.K., 2018. Deep learning with applications using Python. http://dx.doi.org/10.1007/978-1-4842-3516-4.

Ndikumana, E., Minh, D.H.T., Baghdadi, N., Courault, D., Hossard, L., 2018. Deep recurrent neural network for agricultural classification using multitemporal SAR sentinel-1 for Camargue, France. Remote Sens. 10, http://dx.doi.org/10.3390/rs10081217.

Patel, M.M., Tanwar, S., Gupta, R., Kumar, N., 2020. A deep learning-based cryptocurrency price prediction scheme for financial institutions. J. Inf. Secur. Appl. 55, http://dx.doi.org/10.1016/j.jisa.2020.102583.

Patil, K., Deo, M.C., 2017. Prediction of daily sea surface temperature using efficient neural networks. Ocean Dyn. 67, 357–368. http://dx.doi.org/10.1007/s10236-017-1032-9.

Persio, L., Di, O., 2017. Recurrent neural networks approach to the financial forecast of Google assets. Int. J. Math. Comput. Simul. 11, 7–13.

Sarkar, P.P., Janardhan, P., Roy, P., 2020. Prediction of sea surface temperatures using deep learning neural networks. SN Appl. Sci. 2, 1–14. http://dx.doi.org/10.1007/s42452-020-03239-3.

Sezer, O.B., Ozbayoglu, A.M., 2018. Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. Appl. Soft Comput. 70, 525–538. http://dx.doi.org/10.1016/j.asoc.2018.04.024.

Shao, Q., Li, W., Han, G., Hou, G., Liu, S., Gong, Y., Qu, P., 2021. A deep learning model for forecasting Sea surface height anomalies and temperatures in the South China Sea. J. Geophys. Res. Ocean 126, 1–18. http://dx.doi.org/10.1029/2021JC017515.

Shao, Z., Zhang, Z., Wang, F., Xu, Y., 2022. Pre-training enhanced spatial–temporal graph neural network for multivariate time series forecasting. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022. Association for Computing Machinery, Washington, DC, USA, http://dx.doi.org/10.1145/3534678.3539396.

Shen, C., 2018. A transdisciplinary review of deep learning research and its relevance for water resources scientists. Water Resour. Res. 54, 8558–8593. http://dx.doi.org/10.1029/2018WR022643.

Soydaner, D., 2020. A comparison of optimization algorithms for deep learning. Int. J. Pattern Recognit. Artif. Intell. 34. http://dx.doi.org/10.1142/S0218001420520138.

Sun, X., Zhou, F., Dong, J., Gao, F., Mu, Q., Wang, X., 2017. Encoding spectral and spatial context information for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 14, 2250–2254. http://dx.doi.org/10.1109/LGRS.2017.2759168.

Tharwat, A., 2019. Parameter investigation of support vector machine classifier with kernel functions. Knowl. Inf. Syst. 61, 1269–1302. http://dx.doi.org/10.1007/s10115-019-01335-4.

Tölö, E., 2020. Predicting systemic financial crises with recurrent neural networks. J. Financ. Stab. 49, http://dx.doi.org/10.1016/j.jfs.2020.100746.

Wang, L., He, Y., 2022. M2STAN: Multi-modal multi-task spatiotemporal attention network for multi-location ultra-short-term wind power multi-step predictions. Appl. Energy 324, 119672. http://dx.doi.org/10.1016/j.apenergy.2022.119672.

Wang, Q., Meng, Z., Li, X., 2017. Locality adaptive discriminant analysis for spectral-spatial classification of hyperspectral images. IEEE Geosci. Remote Sens. Lett. 14, 2077–2081. http://dx.doi.org/10.1109/LGRS.2017.2751559.

Xiao, C., Chen, N., Hu, C., Wang, K., Gong, J., Chen, Z., 2019. Short and mid-term sea surface temperature prediction using time-series satellite data and LSTM-AdaBoost combination approach. Remote Sens. Environ. 233, 111358. http://dx.doi.org/10.1016/j.rse.2019.111358.

Xie, J., Zhang, J., Yu, J., Xu, L., 2020. An adaptive scale sea surface temperature predicting method based on deep learning with attention mechanism. 17, pp. 740–744.

Xue, Y., Leetmaa, A., Ji, M., 2000. ENSO prediction with Markov models: The impact of sea level. J. Clim. 13, 849–871. http://dx.doi.org/10.1175/1520-0442(2000)013<0849:EPWMMT>2.0.CO;2.

Yang, Y., Dong, J., Sun, X., Lima, E., Mu, Q., Wang, X., 2018. A CFCC-lstm model for sea surface temperature prediction. IEEE Geosci. Remote Sens. Lett. 15, 207–211. http://dx.doi.org/10.1109/LGRS.2017.2780843.

Yao, S.L., Luo, J.J., Huang, G., Wang, P., 2017. Distinct global warming rates tied to multiple ocean surface temperature changes. Nat. Clim. Chang. 7, 486–491. http://dx.doi.org/10.1038/nclimate3304.

Yu, X., Shi, S., Xu, L., Liu, Y., Miao, Q., Sun, M., 2020. A novel method for sea surface temperature prediction based on deep learning. Math. Probl. Eng. 2020, http://dx.doi.org/10.1155/2020/6387173.

Zhang, Z., Pan, X., Jiang, T., Sui, B., Liu, C., Sun, W., 2020. Monthly and quarterly sea surface temperature prediction based on gated recurrent unit neural network. J. Mar. Sci. Eng. 8, http://dx.doi.org/10.3390/JMSE8040249.

Zhang, Q., Wang, H., Dong, J., Zhong, G., Sun, X., 2017. Prediction of sea surface temperature using long short-term memory. IEEE Geosci. Remote Sens. Lett. 14, 1745–1749. http://dx.doi.org/10.1109/LGRS.2017.2733548.