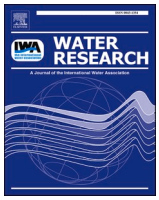
Water Research 287 (2025) 124283

Contents lists available at ScienceDirect

Water Research

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



Shallow learning model for long-term cyanobacterial bloom forecasting in real-time monitoring system

Juan Sandubete-López, Raul Fernandez-Fernandez *, José A. Lopez-Orozco, José L. Risco-Martín

Universidad Complutense de Madrid (UCM), Plaza de las Ciencias, 1, Madrid 28040, Spain

ARTICLE INFO

Keywords:
Bloom forecasting
LSTM networks
Microcontroller
Quantized networks

ABSTRACT

The success of Deep Learning (DL) methods in recent years has popularized complex architectures with extensive layers and parameters, enabling models to capture intricate relationships and extract relevant hidden features. While these architectures achieve impressive results in many classical applications, they are often prone to overfitting and are too costly to be implemented for edge-computing applications, particularly in real-time series forecasting tasks. This paper introduces a shallow Long-Short Term Memory (LSTM) neural network model capable of forecasting cyanobacterial blooms with up to a 70 % accuracy for a 28-day time horizon. This model is embedded in a micro controller unit after applying a quantization process. Unlike traditional methods that rely on centralized processing, our edge-based approach offers real-time, on-site forecasting capabilities, reducing latency and dependency on external infrastructure. We propose it as a cost-effective, low-power and easy to implement edge-based AI system for monitoring buoys, capable of broadcasting predictions and raw measurements through wireless communication. The performance of our model is evaluated with respect to state-of-the-art models and results are obtained for four forecasting horizons (16, 20, 24, 28 days) using Mean Absolute Percentage Error (MAPE). It shows to be 10 points more accurate than the other considered models for the worst-case scenario. The proposed system can be used to aid human forecasting experts or as a standalone system.

1. Introduction

Cyanobacteria blooms occur when there is an exponential growth of cyanobacteria in aquatic systems, affecting the water quality of water reservoirs and threatening with dangerous effects on human health (Huisman et al., 2018). Often produced as a consequence of eutrophication, these blooms affect the aquatic ecosystem by producing toxins that directly affect other aquatic species, birds and terrestrial animals (Turley et al., 2022). Even if when no toxins are released, the death of massive amounts of cyanobacteria depletes the dissolved oxygen. This effect, for example, produced the environmental collapse of the Mar Menor, one of the largest Mediterranean coastal lagoons (Sandonnini et al., 2021). The introduction of mitigation and control measures has proven to successfully reduce cyanobacterial blooms to safety levels (Ibelings et al., 2016). Early warning systems include technologies, like remote sensing, where satellites are used to detect algae or cyanobacterial tides through Synthetic Aperture Radar (SAR) or optical imaging. These technologies usually provide low sampling rates and can nevertheless be affected by factors like the weather or the turbidity of the water, which can be challenging for fine modelling of blooms dynamics (Zahir et al., 2024). *In-situ* detection systems have also been proposed based on flow imaging instruments or chemical measurements (Barrowman et al., 2024). Although these are precise detection systems, they usually require a high initial investment and continuous maintenance (Zahir et al., 2024). Additionally, control measures usually depend on human experts who guess when the next bloom will occur via visual inspection of the situation or the available information. As a consequence, any human error can have a critical impact in the correct implementation of these preventive measures. In this paper, we propose a framework to develop an autonomous monitoring system that can forecast cyanobacteria levels in real-time with a 70 % accuracy within a 28-day time horizon. This system can be used to aid human experts, for example, by triggering alerts when a relevant growth trend is detected or by supporting humans' decisions, providing the experts with visualizations of potential future situations. It could also be used as a standalone system for the application of automatized preventive measures. Current

* Corresponding author.
E-mail address: raufer06@ucm.es (R. Fernandez-Fernandez).

https://doi.org/10.1016/j.watres.2025.124283
Received 23 January 2025; Received in revised form 21 July 2025; Accepted 22 July 2025
Available online 23 July 2025

centralized systems often suffer from high latency and require significant infrastructure, having associated challenges for remote or resource-constrained environments (Zahir et al. 2024). Our edge-based approach addresses these challenges by enabling real-time processing directly at the monitoring site. The proposed model is designed to be deployed within a low-power and cost-effective Micro-Controller Unit (MCU). The result is a low-cost system that can be easily deployed in any mass of water without requiring any additional infrastructure after a fine-tuning step to adapt to the local dynamics. This system provides predictions that can be used to introduce mitigation and control measures.

To develop this system, we apply recent advances in Deep Learning (DL) techniques, like dropout or batch normalization for regularization, within a shallow neural network. DL frameworks introduce complex architectures that have been successfully applied to many problems in a wide range of topics (ChatGPT, 2022; Fernandez-Fernandez et al., 2023; Gatys et al., 2016). At the same time, however, the overfitting problem is inherent to neural networks with complex architectures and many parameters (Nielsen, 2015). This problem is aggravated in contexts where DL methods are not capable of finding meaningful solutions. One of these complex problems is real-time series forecasting (Masini et al., 2023) which is the method required to perform cyanobacteria level forecasting. Following this, in this work, the introduction of a shallow neural network is proposed with two different goals: first, we avoid the presented overfitting problem when introducing complex architectures with time series forecasting, and second, we propose a model light enough to be embedded within an MCU. To improve the performance of this shallow neural network architecture, we will make use of the different methods that have been proposed as the result of the success of Deep Learning architectures to improve our shallow neural network performance.

For training the architecture, we use real high-frequency samples obtained from a multiparametric probe mounted on a water column profiler within the Cuerda del Pozo reservoir in Soria, Spain. Multiple samples were obtained every hour for a period of five years, from 2010 to 2015, at multiple depths with periods of missing data. A preprocessing step was introduced to improve this data: data sampling irregularity was removed by grouping samples into 5 min-average samples, gaps were filled with the last available value, and values were normalized into a range [−1.0, 1.0]. Augmentation techniques, as explained in Section 3, were also applied to the resulting adapted data before training the model. Phycocyanin readings were used as the proxy for cyanobacteria (Ahn et al., 2023). Temperature, phycocyanin, and date values are the parameters used as inputs for the model. The outputs of the model are 4 phycocyanin values corresponding to the 4 predicted time horizons (16, 20, 24 and 28 days). The data of these three parameters is transmitted to a MCU through serial communication, where they are inserted in a FIFO memory buffer that feeds the quantized model to make edge forecasting. The final framework is depicted at Fig. 1 and can be summarized with the following steps: First, a high frequency data extraction is performed using a multiparametric probe; then, a shallow neural network is trained using this data; this network is quantized to fit the MCU; finally, the resulting model can be used to predict the readings of the probe up to the 28 following days with a 70 % accuracy.

The proposed architecture introduces an LSTM layer with a ReLU activation as the only hidden layer to take advantage of the loop of information defined by this type of Recurrent Neural Networks (RNN, (Bengio et al., 1994)). A batch normalization layer (Ioffe and Szegedy, 2015) is used to improve training stabilization and a dropout layer (Srivastava et al., 2014) is applied to further reduce overfitting. A dynamic-range post-training quantization (Han et al., 2015) is applied to the resulting model converting the 32-bit floating-point values of the model into 8-bit integers. This further reduces its memory footprint and the required computation making it suitable for the MCU, while, at the same time, improving its inference rate and energy consumption (Ray, 2022).

The remaining of this paper is organized as follows. Section 2 presents an introduction to the problem of Bloom Forecasting and related DL architecture. Section 3 defines the dataset and describes how the data was obtained, Section 4 introduces the architecture, the training parameters and the quantization process, and Section 5 shows the proposed hardware design and the real-time application procedure. Section 6 contains the results obtained with the proposed framework. Finally, Section 7 and Section 8 discuss the results and propose some conclusions and ideas resulting from this paper, including future related lines of research.

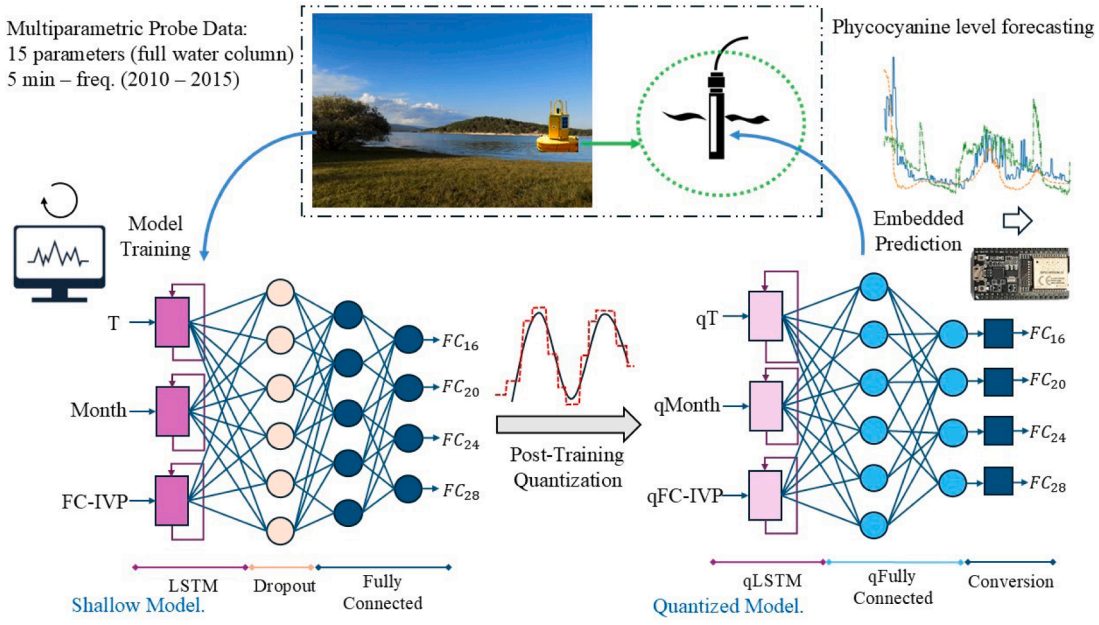


Fig. 1. Conceptual schema of the proposed framework.

2. Bloom forecasting and monitoring systems

Classical cyanobacteria bloom forecasting methods have relied on statistical models and heuristic approaches in the past. Several of these models often involve the analysis of environmental variables, including nutrients and weather conditions, to predict the bloom behavior (Dignum et al., 2005; Huisman and Hulot, 2005; Smith, 2003). While these approaches provide valuable insights into bloom dynamics, they usually require an expert for the interpretation of the data and the assumption of linear or straightforward relationships between variables. The low confidence levels of these methods only allow the implementation of general preventive or mitigation measures. Recently, the introduction of more advanced (ML) techniques, like Random Forest or Support Vector Machines (Ribeiro and Torgo, 2008; Yajima and Derot, 2018), have allowed the introduction of models capable of capturing nonlinear relationships for blooms prediction. In theory, some of these models provide long-term forecasts that would enable the introduction of relevant preventive measures. However, the majority of the models were not good enough to be used without an expert to make meaningful decisions (Cruz et al., 2021).

In recent years, DL techniques with specialized architectures, like RNN, have excelled at extracting complex relationships within large datasets of time series (Hewamalage et al., 2021). In particular, applying LSTM networks has been the most promising branch within RNN (Chimmula and Zhang, 2020). Some works combining LSTM, Fully Connected layers, and bloom forecasting have also been proposed (Cho and Park, 2019; Liu et al., 2022; Zheng et al., 2021). However, these models were limited to offering accurate short-term forecasts spanning only a few days (1–3 days). This forecast time is in line with the growing duplication times of cyanobacteria (Chorus and Welker, 2021). This implies that although they provide relevant information about short term evolution, these models are not able to encode long-term growing dynamics.

Regarding bloom monitoring systems, conventional field methods involve periodic on-site sampling requiring a following laboratory analysis (Kang et al., 2020; You et al., 2022) using a visual detection with a microscopy inspection as a common approach (Aguilera et al., 2023). Other real-time monitoring systems allow for continuous onsite inspection of the blooms. These systems can be classified as a function of the measure technique performed: based on fluorescence measurement (Izydorczyk et al., 2009; Richardson et al., 2010); based on electro-chemical biosensors (Pescheck et al., 2022); or flow-cytometry sensors (Ruiz-Villarreal et al., 2022). Some monitoring systems have also introduced ML and DL advanced methods (Xiao et al., 2024). These methods, however, are usually run in computers or servers after gathering the data from the sensing systems (Zahir et al., 2024). This increases the associated costs and slows down the monitoring process. Edge computing allows for immediate data processing and decision-making at the source, which is crucial for timely interventions in bloom management. Edge processing does not only reduce the need for data transmission to central servers but also enhances the system's resilience and scalability. This approach was followed by Yuan et al. (2023), who presented an edge AI-chip for real-time classification of algae from imaging flow-cytometry. More generally, edge AI has been applied to the field of environmental monitoring for applications such as landslide prediction and detection or for air pollution monitoring (Joshi et al., 2024). This field shows an increasing interest in distributed and real-time sensing applications.

The design of monitoring systems with embedded forecasting models that do not require additional infrastructure allows the creation of distributed cost effective and autonomous solutions. However, the design of these systems has some challenges associated. The low-power and average processing speed of an MCU requires a specific adaptation of the DL models (Ray, 2022). The most extended technique in these cases consists of performing a quantization. This technique performs the conversion of the data types of a DL model, usually from 32-bits floating-point, to lower bit data types by studying the range of the values during the inference of some input sample data (Han et al., 2015). This reduces the size of the model and enhances its inference latency (Wu et al., 2020). There are several quantization techniques defined within the state of the art as dynamic-range Post-Training Quantization (PTQ) (Han et al., 2015), Quantization-Aware Training (QAT) (Jacob et al., 2018), or Hardware-Aware Quantization (HAQ) (Rusci et al., 2020), being PTQ the simplest and most widely used (Ray, 2022).

As a conclusion, the application of DL techniques to cyanobacteria bloom forecasting, although promising, presents several challenges. The inherent complexity of bloom dynamics and the lack of high-quality training data make overfitting a recurrent problem within these models. Applications should focus on the development of robust, simple, and generalizable models that are easier to understand and that can be applied to real-world scenarios with diverse environmental conditions. At the same time, monitoring systems commonly require from on-site data gathering and additional infrastructure to process the extracted data. These solutions should have a high degree of autonomy, be scalable, cost effective, and easily deployed in areas often distant from population centers. The system proposed in this work was designed taking these considerations in account.

3. Data description

The data used in this paper was obtained from the water reservoir Cuerda del Pozo, Soria, Spain. This multipurpose water reservoir is used for water supply, recreation, irrigation, and energy production. Measures were obtained across multiple depths using a multiparametric probe mounted in a floating buoy during five years, from April 2010 to September 2015. A total of 87,684 measures across 15 different parameters were obtained. The available parameters are described in Table 1.

This data was studied and preprocessed before being introduced to the model. In order to carry out this study, phycocyanin was used as a more reliable proxy for the presence of cyanobacteria than chlorophyll a (Ahn et al., 2011). First, a correlation study (summarized in Fig. 2) between all parameters was performed. For groups of highly correlated parameters (correlation greater than 75 %), only one of the group parameters was selected for further study. These parameters, selected after this correlation study, are depicted in the completeness graph presented in Fig. 3. This completeness graph shows the number of measures in which all of these parameters were measured as a function of the total number of measures and the time period. It also shows that the sample

Table 1
Parameter dictionary. The MSE value, obtained via input noise injection, is a measure of how important the input for an initial test network is. Higher MSE values imply a higher weight of the input for the network.

Acronym	Description	MSE value
D	Depth	0
T	Temperature	0.8
C	Electric Conductivity	0.9
C25	Temperature Normalized Electric Conductivity	0.9
DO	Dissolved Oxygen	0.4
DOSAT	Saturated Dissolved Oxygen	–
pH	pH measure	0.8
ORP	Oxidation-Reduction Potential	0.3
FC_IVF	Phycocyanin Fluorescence	1.2
CIANO_EQ	Equivalent Biovolume of Cyanobacteria	–
CYNQ	Equivalent Concentration of Cyanobacteria	–
CHLA_IVF	Chlorophyll a Fluorescence	0.3
CHLA_LEQ	Chlorophyll a Equivalent Extracted in Lab	–
CDOM	Chromophoric Dissolved Organic Matter Fluorescence	0.7
CDOMT	Temperature Normalized CDOM	–

	Month	Year	D	T	C	C25	DO	DOSAT	PH	ORP	FC_IVF	CIANO_EQ	CYNQ	CHLA_IVF	CHLA_LEQ	CDOM	CDOMT
Month	1.00	-0.28	0.03	0.29	0.47	0.57	-0.32	-0.23	0.02	-0.24	0.15	0.19	0.19	-0.16	-0.13	-0.27	-0.17
Year	-0.28	1.00	0.14	-0.25	-0.22	-0.08	0.21	0.12	-0.14	0.48	-0.08	-0.24	-0.24	0.05	0.39	0.15	0.25
D	0.03	0.14	1.00	-0.28	-0.18	0.06	-0.51	-0.64	-0.41	-0.01	-0.19	-0.13	-0.13	-0.35	-0.17	0.22	0.18
T	0.29	-0.25	-0.28	1.00	0.87	0.31	-0.38	-0.01	0.31	-0.31	0.06	0.14	0.14	-0.01	-0.16	-0.44	-0.24
C	0.47	-0.22	-0.18	0.87	1.00	0.70	-0.46	-0.15	0.28	-0.30	-0.05	0.03	0.03	-0.06	-0.15	-0.55	-0.34
C25	0.57	-0.08	0.06	0.31	0.70	1.00	-0.39	-0.31	0.12	-0.12	-0.15	-0.12	-0.12	-0.10	-0.06	-0.49	-0.35
DO	-0.32	0.21	-0.51	-0.38	-0.46	-0.39	1.00	0.93	0.36	0.30	0.19	0.05	0.05	0.43	0.34	0.08	-0.01
DOSAT	-0.23	0.12	-0.64	-0.01	-0.15	-0.31	0.93	1.00	0.50	0.19	0.24	0.11	0.11	0.45	0.31	-0.07	-0.09
PH	0.02	-0.14	-0.41	0.31	0.28	0.12	0.36	0.50	1.00	-0.07	0.05	0.04	0.04	0.26	0.08	-0.31	-0.25
ORP	-0.24	0.48	-0.01	-0.31	-0.30	-0.12	0.30	0.19	-0.07	1.00	-0.26	-0.39	-0.39	0.08	0.21	-0.07	-0.04
FC_IVF	0.15	-0.08	-0.19	0.06	-0.05	-0.15	0.19	0.24	0.05	-0.26	1.00	0.95	0.95	-0.04	-0.11	0.05	-0.12
CIANO_EQ	0.19	-0.24	-0.13	0.14	0.03	-0.12	0.05	0.11	0.04	-0.39	0.95	1.00	1.00	-0.09	-0.18	-0.00	-0.17
CYNQ	0.19	-0.24	-0.13	0.14	0.03	-0.12	0.05	0.11	0.04	-0.39	0.95	1.00	1.00	-0.09	-0.18	-0.00	-0.17
CHLA_IVF	-0.16	0.05	-0.35	-0.01	-0.06	-0.10	0.43	0.45	0.26	0.08	-0.04	-0.09	-0.09	1.00	0.73	0.13	0.09
CHLA_LEQ	-0.13	0.39	-0.17	-0.16	-0.15	-0.06	0.34	0.31	0.08	0.21	-0.11	-0.18	-0.18	0.73	1.00	0.14	0.25
CDOM	-0.27	0.15	0.22	-0.44	-0.55	-0.49	0.08	-0.07	-0.31	-0.07	0.05	-0.00	-0.00	0.13	0.14	1.00	0.96
CDOMT	-0.17	0.25	0.18	-0.24	-0.34	-0.35	-0.01	-0.09	-0.25	-0.04	-0.12	-0.17	-0.17	0.09	0.25	0.96	1.00

Fig. 2. Correlation values between all parameters available in the dataset. Dark blue colors correspond to lower correlation values while dark red ones correspond to a higher correlation.
Parameters: [D, T, C, C25, DO, pH, ORP, FC_IVF, CHLA_IVF, CDOMT]

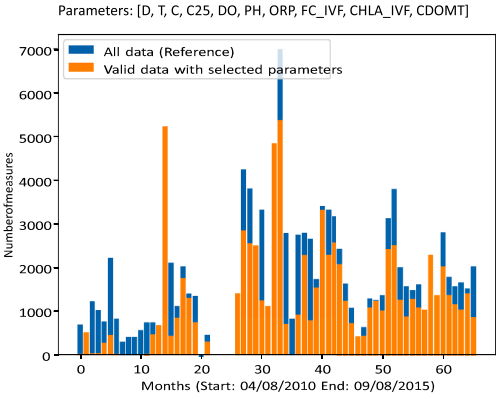


Fig. 3. Portion of measures where all the selected parameters had available samples in the dataset.

frequency was not homogeneous between all time periods.

Using this dataset, a study of the most relevant parameters via input noise injection (Grandvalet et al., 1997) was performed to reduce further the number of input parameters. The results are depicted in the third column of Table 1. These values are the average Mean Square Error (MSE) obtained when introducing noise to the measured parameter. A higher MSE depicts a higher dependence of the network on the parameter. This study was performed using an initial LSTM architecture that was later fine-tuned to the final parameters proposed in this paper. Following these results and the results obtained in this previous paper (Fournier et al., 2024) only the Temperature (T) parameter, the date (date) information, and the phycocyanin measures (FC_IVF) were used as inputs for the network. The rest of the parameters were not used and are not required by the system. Incomplete data was interpolated using a linear interpolation. The resulting dataset contains a value for each of these three parameters with a 5-minute frequency for five years. Higher frequency measures in the dataset were averaged to this 5-minute frequency.

For training the model, a dataset of couples of input/output time series was generated for the training and testing steps. Data was normalized to generate both datasets by removing the mean and scaling to unit variance. The training dataset was defined from April 2010 to May 2014, while the test dataset was defined with the rest of the data, from May 2014 to September 2015. A total of 30 input values were introduced to the network. Each of these values is the maximum of a 4 days (96 h) rolling window starting with the prediction day and spanning to the previous 4 months. A total of 4 outputs are obtained from the network. These four outputs correspond to the four selected forecasting horizons which are the next 16, 20, 24 and 28 days starting with the current day. Each of these values is the expected maximum phycocyanin level for the selected 4 days span (i.e. the 4-day horizon predicts the maximum phycocyanin value for the next 4 days, the 8-day horizon between days 4 and 8, and so on). Note that the days here are defined as 24-hour windows and not natural days. From the total 34,560 values, 1152 values were used to compute each of the inputs. One input/output couple was generated from each hour of the dataset. Predictions in this paper are obtained using a rolling forecast approach (Liu et al., 2021).

Using the generated training dataset, a data augmentation step was performed to multiply the available data by nine. This is the equivalent of introducing to the dataset 8 new datasets. The first three datasets were respectively generated with a jitter operation with standard deviation values of 0.03, 0.1, and 0.2. The two following datasets were the results of performing a scale transformation with standard deviation values of 0.6 and 1. The next two datasets were generated with a magnitude warp operation with standard deviation values 1 (with 4 knots) and 0.6 (with 14 knots). Finally, a time-warping operation was introduced to generate the last two datasets with standard deviation values of 0.4 (with 4 knots) and 0.2 (with 14 knots). All data augmentation operations were performed with the code by Iwana et al. (Iwana and Uchida, 2021).

4. Model architecture and training

The proposed model is a shallow LSTM architecture with one LSTM layer (Hochreiter and Schmidhuber, 1997) and a Fully Connected layer as hidden layers. This is an approach broadly followed in literature which usually performs well and easily converges for time-series forecasting problems (Lim and Zohren, 2021). Several improvements and tuning iterations were made to this basic architecture to improve performance. First, a dropout layer (rate=0.3) was added to improve the generalization capability of the network to adapt to different contexts and environments. Adam (Kingma and Ba, 2015) was selected as the optimization algorithm since it is broadly used, showing good results (Hajji et al., 2024) in many different applications and a ReLU (Nair and

Hinton, 2010) activation function was selected for each layer. The MSE was selected as the loss function of the network. Finally, an early stopping condition was added to avoid overfitting. This condition is triggered when five consequtive iterations are performed during training without an improvement in the validation loss.

In order to validate this model, we compared the results obtained with this architecture with two state-of-the-art models for bloom forecasting using LSTM networks. The first model, the Deep LSTM model (Zheng et al., 2021), proposes a Deep Learning architecture with three hidden LSTM layers and 256 neurons in each of the layers. This is a standard deep learning setup with a considerable number of parameters. The second model, the CNN-LSTM model (Cao et al., 2022) introduces three LSTM and three hidden convolutional layers, reducing the number of neurons to 3 in each of the layers. This architecture reduces the complexity of each of the layers while introducing three new ones. Both of the presented architectures present a higher number of parameters and a higher complexity than the shallow LSTM neural network proposed in this paper.

The selected architecture for our Shallow LSTM network is composed of an input layer, a hidden layer, and the output layer. The hidden layer is a Long Short-Term Memory (LSTM) layer (Hochreiter and Schmidhuber, 1997) with 7 units and a ReLU activation function. To this layer, a dropout (with rate=0.2) and a batch normalization step were added to improve network's generalization and training convergence, respectively. The learning rate was fixed to 5×10^{-6}, the batch size was set to 100, and the loss function was MSE. The training was executed until reaching 1000 iterations or 30 consecutive iterations without increasing validation accuracy. Training parameters for the three models are defined in Table 2.

The training was performed using a HP Victus Laptop with 16GB of RAM and an 11th Generation Intel Core i7–11800H processor with a base frequency of 2.30 GHz and 16 cores. The graphical card of the laptop was a NVIDIA Corporation GA106M graphical card, specifically the GeForce RTX 3060 Mobile / Max-Q variant. The operating system utilized for training was Ubuntu 22.04 LTS 64-bit edition. The average time, over 7 different execution, taken to train each of the models is depicted in Table 3. Here, the high standard deviation value obtained with the proposed model with respect to the other two is the result of not defining a fixed number of steps but performing an early stop technique. A maximum of 100 iterations were set for the Deep LSTM and CNN-LSTM networks. The proposed model did not use a fixed maximum number of iterations but always required less iterations than this value. Training time differences come from different architectural complexities. For instance, the proposed Shallow LSTM model and the CNN-model have around 400 trainable parameters, while the Deep LSTM model has around 1.3 million trainable parameters.

In terms of training error, this is the forecasting error introduced by the network when predicting values used for training, Table 2 depicts the training results of the three models. The Deep LSTM model has the best performance. In terms of validation loss, this is the forecasting error introduced by the network when predicting values not used for training,

Table 2
Architectural parameters.

Model	Layers (Neurons, Kernel)	Learning rate	Batch size	Train loss	Val. loss
Proposed Shallow LSTM	LSTM(7)-Dropout-FC(7)	0.0001	5	1.23	2.97
Deep LSTM (Zheng et al., 2021)	LSTM(256)-LSTM(256)-LSTM(256)	0.0001	32	0.06	3.85
CNN-LSTM (Cao et al., 2022)	CNN(3,5)-CNN(3,5)-CNN(3,5)-LSTM(3)-LSTM(3)-LSTM(3)	0.01	32	0.73	6.1

Table 3
Time information for the training.

Model	Time [s]	Standard Deviation [s]
Proposed Shallow LSTM	30	13.5
Deep LSTM	1283	7.56
CNN-LSTM	13	0.32

the proposed Shallow LSTM network achieves the best results. From these two parameters, the validation loss is the most important in terms of measuring how good the real performance of the model is. A low training error and a high validation error usually mean a model that has overfitted the data, as in the Deep LSTM and CNN-LSTM models. These results, however, have to be considered given that the Deep LSTM and CNN-LSTM architectures were proposed within a different context, using different data and with different prediction horizons. The results here obtained, do not invalidate the results of the original papers.

Even though the resulting trained shallow learning model does not require a big amount of memory space – around 160 KB –, it is composed of floating-point parameters that reduce the inference latency of the model and increase its power consumption (Ray, 2022). Thus, the PTQ technique is applied to convert the 32-bit floating point parameters into 8-bit integer parameters. For this process, 900 samples from the training data are used. While the inputs are quantized into 8-bit values following the process described in Section 5, the outputs are automatically converted into floating point data by a conversion module created during the quantization of the model. All the code used in this paper has been open-sourced and is available online.

5. Embedded implementation

The problem tackled in this work is not limited to the prediction of the bloom's evolution over time. An additional contribution is the development of an add-on edge computing unit for algal monitoring systems composed of an independent power unit and a MCU to forecast the dynamics of the cyanobacteria bloom. The MCU takes the data from the already available sensors on the blooms monitoring system. Thus, the layout of the considered problem also includes the needs for the right operation of the system, namely: communication with the sensors of the buoy, the signal processing, the quantization of the signals from the sensors, and the management of the historical data (updating the registers).

The MCU model chosen, shown in Fig. 4, to serve as forecasting unit is an ESP32-WROOM-32. This model has a relatively big integrated storage capacity (4 MB of FLASH memory), medium-high computation capabilities (dual-core, up to 240 MHz), integrated wireless connectivity (Wi-Fi and Bluetooth), and a budget-friendly option. This device has a limited RAM capacity, which is an obstacle for storing the input data from the last 4 months. To overcome this problem, the data is quantized

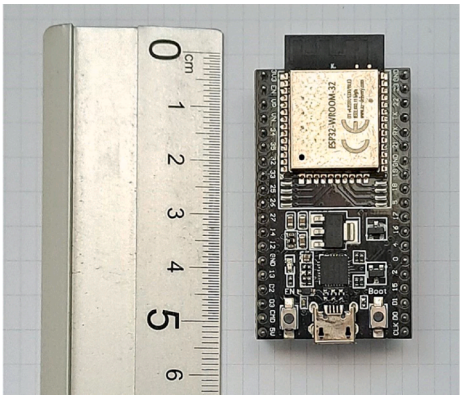


Fig. 4. ESP32 microcontroller development board, used for this work.

into 8-bit values and it's directly managed at the FLASH memory level. The MCU is implemented to periodically gather data from the sensors with a 5 min frequency. This data can be transferred through serial communication from the monitoring buoy to the MCU containing the proposed shallow neural network model that will be in charge of performing the forecasting. Fig. 7 depicts a high-level implementation of the discussed system. It is assumed that the monitoring buoy has internally a device like a MCU with capability to send out, for example through serial communication or any other method, the raw data from the sensors. Due to the RAM limitations of the MCU, in order to run inferences of the DL model, as discussed in the previous Section 4, it is required to implement the quantized DL model instead of the standard version. The inputs of this quantized version are integers in a range from −128 to 127, in contrast to the data coming from the sensors, which are floating point numbers with a higher degree of variability. Thus, to transform the floating-point values into the required discrete range, the following equation can be applied:

$$Si,\mathrm{int} = \left\lfloor (Si,\mathrm{float} - Si,\mathrm{min}) \frac{255}{Si,\mathrm{max} - Si,\mathrm{min}} - 128 \right\rceil \quad (1)$$

where the operator $\lfloor\rceil$ represents the round operation to the closer integer value $Si,float$ is the floating point value from the i th sensor, and (Si,min, Si,max) are the minimum and maximum value that $Si,float$ can have.

The integer values of each sensor are then inserted into a circular buffer containing 103,680 cells, corresponding to 34,560 samples of the three sensors of interest, collected each 5 min during four months. The circular register works as a First-In-First-Out (FIFO) register: each time a new set of samples arrives, the oldest set is replaced by the new one. If, for a sampling period, the sensors do not provide the MCU with a measurement, the last registered value is inserted again, filling the gap. Using these measurements, the maximum values for each parameter (phycocyanin and temperature) are obtained by grouping the data of the register in blocks of 96 h, four days, following the format required for the input model.

The forecasting unit then runs the inference and publishes both the original floating-point data from the sensors and the prediction of evolution of the bloom, for the four time-horizons, to a MQTT distributed network over a Wi-Fi connection. The user can connect to this network with any compatible device, receiving the desired data every 5 min from the defined topics. The edge-based design significantly lowers operational costs by minimizing data transmission and processing requirements. Its compact and autonomous nature makes it ideal for deployment in remote locations without the need for extensive infrastructure. This allows the user to have an online live stream of real data and forecasting data from the network that summarizes the current state of the measured parameters and the predicted state. The edge-based implementation not only maintains high forecasting accuracy but also ensures that predictions are made in real-time, providing immediate insights that are critical for effective bloom management.

6. Results

Results were obtained for the four different time horizons described in Section 3. Each of these four time horizons correspond to a different output of the network. Only one model was trained for all time horizons. Values were denormalized for the study of the results.

In order to validate the performance of the proposed architecture, three different initial experiments were performed, one for each studied architecture. The Mean Absolute Percentage Error (MAPE) was used as the evaluation metric for all of the experiments.

Results are depicted in Table 4. The first experiment, corresponding to the results labeled as MAPE S (Shallow) in the table, takes the continuous values predicted by the proposed network and directly compares them to the values in the testing dataset. Hence, it evaluates the performance of the network as it is designed, i.e. to predict the continuous value of phycocyanin after a given period of time (16, 20, 24 or 28 days ahead). The second and third experiments are evaluations of successful models found in literature for time-series prediction, under the same conditions as the proposed shallow model, labeled: MAPE Deep LSTM (Zheng et al., 2021) and MAPE CNN-LSTM (Cao et al., 2022). These evaluations are made to validate the use of the proposed shallow model instead of a deep learning model. Fig. 5 depicts a graph with the accuracy values for each of the experiments. This values are the complementary values of the MAPE values obtained in Table 4. As stated in Section 4 both the Deep LSTM and the CNN-LSTM had overfitting behavior during training.

Table 4
Experimental results for the test dataset [Error, %]. MAPE was the metric selected for these experiments. The three different models are compared. MAPE S corresponds to the prediction error of the Shallow model, MAPE Deep LSTM for the model by Zheng et al. (2021), and MAPE CNN-LSTM for the model by Cao et al. (2022).

Horizon	16	20	24	28
MAPE Shallow	30.51	30.10	29.14	28.23
MAPE Deep LSTM	43.86	43.51	42.02	41.26
MAPE CNN-LSTM	43.01	43.41	45.76	43.64

This overfitting behavior can also be observed in Fig. 5 for the Deep LSTM model. In this figure, although the proposed shallow LSTM model and the Deep LSTM are both able to predict the first bloom peak with a decent time accuracy, for the rest of the values the proposed shallow LSTM model is faster forecasting the phycocyanin values. This slow forecasting behavior from the Deep LSTM model usually means a dependency from the input data dynamics in terms of forecasting (i.e. the model is not able to completely predict the blooms but starts forecasting a bloom when it is already happening). This behavior usually happens to overfitted models that simply replicate the outputs from the input values.

Obtained results show how the proposed shallow architecture (MAPE Shallow) was the one that achieved the best overall performance. Accuracy values around or higher than 70 % were obtained for all the studied time horizons. Only a 2.5 % accuracy was lost between the first time horizon and the last. In the case of the state-of-the-art deep learning architectures (Deep LSTM, CNN-LSTM), a gap in accuracy is observed with a value higher than 10 % for all time horizons. The shallow model outperforms both deep learning alternatives, which show accuracies lower than a 60 % for the defined experiments.

Once we have validated the use of the proposed shallow model for the forecasting of the bloom dynamics, a post training quantization process was applied to this model as described in Section 5. To emulate the nominal operation of the monitoring system, a hardware-in-the-loop approach was followed, where the computer acted as the secondary MCU shown in Fig. 7. The laptop was in charge of providing the forecast unit with measurements from the test dataset through serial connection every 5 min. To measure the energy consumption of the system, a wattmeter was placed in the power line of the MCU, which produced predictions after every emulated measurement, remaining in standby operation the rest of the time. During inference computing, the MCU had a peak consumption of 1.2 W, in contrast to the 0.75 mW consumed in standby mode. In Table 5 the inference times are shown for the model, and its quantized version, running in the computer used for the training of the model and in the ESP32 microcontroller. The quantized version is 200 times faster than the original model, which allows for its practical implementation within the MCU. Even though it takes 2.23 s, on average, to run inferences in the embedded systems, this implies computing for only 0.75 % of the 5 min interval, which means a low average power consumption of 9.66 mW.

Forecasting results obtained with the quantized model can be observed in Fig. 6. While a degradation in the prediction accuracy is

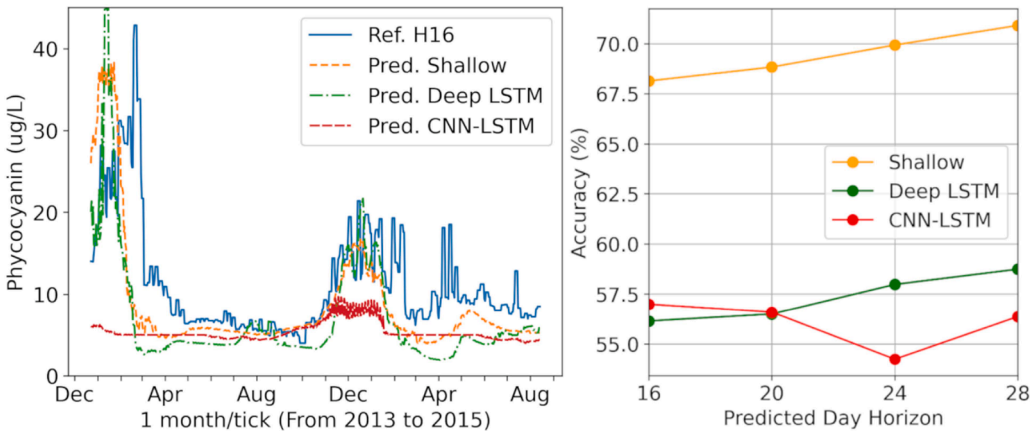


Fig. 5. On the left, predictions for values 16 days ahead are shown as a working example. The ground-truth reference value for the time horizon of 16 days (Ref. H16) is compared to the predictions coming from the presented shallow model and the two deep learning models alternatives. On the right, accuracy values for all the studied time horizons using MAPE.

Table 5
Inference times for the original and quantized (prefix q) models running in personal computer or microcontroller.

Magnitude	Time Mean [s]	Time STD [s]
OP_{PC}	4.4e-2	1.3e-2
qOP_{PC}	2.5e-4	1.4e-3
qOP_{MCU}	2.23	0.44

noticeable after the quantization process, the model still correctly predicts the apparition of phycocyanins' peaks. It is also relevant to note how the predicted peaks are actually faster than the real peaks. This, although it is still a prediction error, also means that the network is not overfitting and cyanobacteria growth is predicted before it actually starts to happen. For the 28-days prediction horizon, Fig. 6.d, although spurious peaks are predicted in May of 2014 and May of 2015, the results remain close to the ground-truth being the less affected output by the quantization of the model.

7. Discussion

Bloom forecasting works are hard to generalize due to the variability between water reservoirs, environmental conditions, and cyanobacteria species. In addition, forecasting natural phenomena is a complex

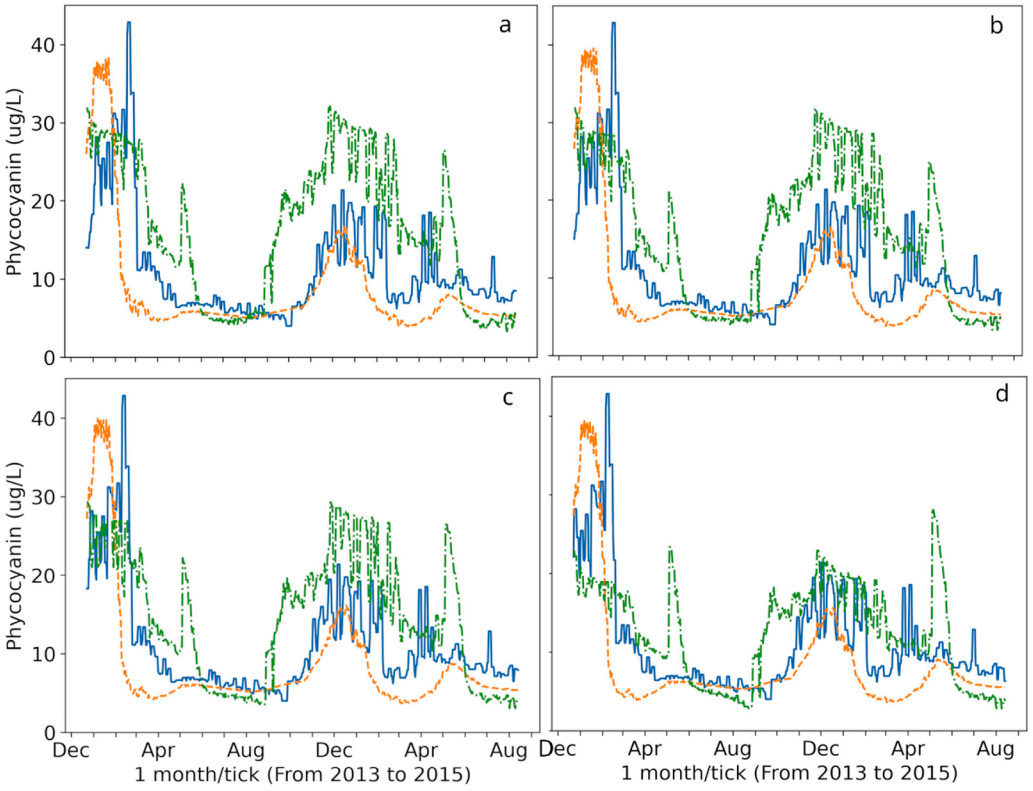


Fig. 6. Experimental results using the proposed framework. Blue data corresponds to the real interpolated data obtained from the probe. Dashed orange data corresponds to the values predicted by the neural network, and dot-dashed green corresponds to the quantized version of the model. Results are depicted for the four different time horizons predicted by the network: a) 16 days ahead, b) 20 days ahead, c) 24 days ahead, and d) 28 days ahead.

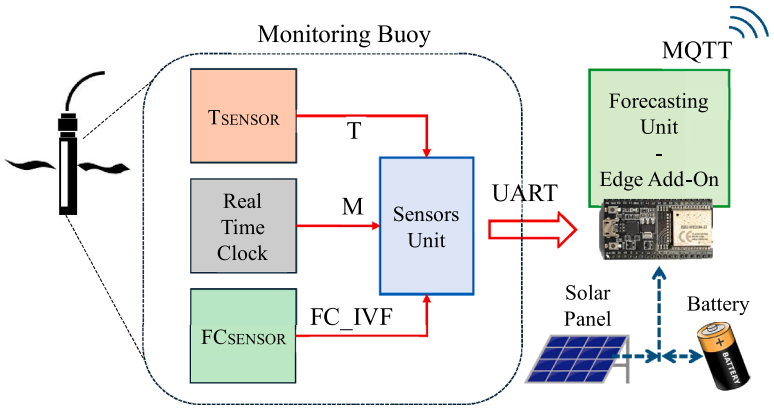


Fig. 7. High level diagram of the designed system. A time sensor, a real-time clock and a fluorescence sensor are connected to an internal microcontroller or device with communication capabilities like Universal Asynchronous Receiver-Transmitter (UART) protocol. This periodically gathers the data (T: Temperature, M: Month, FC_IVF: Phycocyanin In Vivo Fluorescence), with a 5 min frequency, and sends it to the forecasting microcontroller as one packed message. The forecasting microcontroller has wireless connectivity for sending the information to the user.

problem that implies many different variables that are not always available to the system. Introducing complex architectures can lead to overfitting and purely reactive models in this context. Special care must be put on not training a model that just mirrors the last data available. Although DL architectures are able to learn complex and hidden relationships, shallow architectures favor models with simpler tendencies that can give real valuable predictions of the data for complex systems. In addition to this, the low computational cost of shallow architectures makes them feasible for its implementation within edge computing systems. In this paper we proposed the implementation of a shallow neural network architecture within a cost-effective edge computing system.

Many blooms forecast models have been already proposed with diverse results. A representative example is the Auto-Regressive Integrated Moving Average (ARIMA) model presented by Chen et al. (2015). Although this model obtains accurate results to forecast the bloom evolution, its prediction horizon is limited to two days since ARIMA models cannot integrate non-linear patterns. Other models like random forest or support vector machines have a high computational cost and tend to overfit for multivariate time-series (Cruz et al., 2021). Thus, a shallow model based on LSTM layers stands out as an alternative able to represent non-linear relationships, at a moderate computational cost.

The shallow architecture proposed in this paper can make predictions with 70 % accuracy. This accuracy is approximately 10 % higher than the obtained using the alternative state-of-the-art DL architectures. Furthermore, the proposed model achieves medium-term predictions of up to 28 days. This forecasting horizon extends beyond the reach of current state-of-the-art models, such as the models by Zheng et al. (2021) or by Martín-Suazo et al. (2024), which are limited to shorter intervals (up to 196 h). To the best of our knowledge, no existing models in literature provide comparable long-term accuracy. These successful long-term results fit cyanobacteria's real behavior where division times are in the range of days (Chorus and Welker, 2021). Bloom cyanobacterial models should be trained using these time scale horizons to allow the network to model cyanobacteria division behavior.

One important aspect that has to be discussed and considered is that the sample frequency of the data used for training was 5 min. We acknowledge that a 5-minute sample frequency could be a too demanding requirement for some monitoring installations capabilities, budgets or conditions. However, our choice of using a 5-minute frequency was taken with the goal of improving the performance of the model. It is a general consensus within the deep learning and neural network community that increasing the amount of data, along with a well-designed architecture, leads to improvements on the performance of the model (Sun et al., 2017). For instance, in this application, high frequency training data could allow the model to learn high frequency noise filters to remove noise within the input data, improving robustness to sensor variability and environmental fluctuations. More importantly, since the training has already been performed, the proposed model could be fine-tuned to work with any frequency of input data that allows the model to monitor the relevant learned features from temperature and phycocyanin samples. We suspect that this minimum sample frequency will greatly depend on the environment. Future suggested works include a study of how using different sample frequency for prediction impacts the model performance as a function of the environment conditions.

The implementation of the proposed autonomous system is cost-effective and requires minimal infrastructure. Available monitoring systems often necessitate data retrieval from buoys and subsequent processing on a workstation, incurring significant logistical costs. In contrast, our system processes data at the edge, eliminating the need for strong Wi-Fi connections or additional controllers to transmit large amounts of data to fog or cloud IoT layers, which further reduces operational costs. As shown in Table 6, the system's power consumption is capped at 75.0 mW, allowing it to be powered by a small integrated solar panel and a battery. With a standard 2000 mAh battery, the system can operate for approximately 130 h. Integrating a 0.5 W solar panel ensures continuous, autonomous operation. This approach not only minimizes computing costs and resource requirements but also provides a scalable, easily deployable solution for real-time water body monitoring, adaptable to local dynamics without the need for extensive infrastructure. By leveraging edge computing, our system offers a scalable solution that can be easily adapted to various environmental conditions, making it a versatile tool for global water quality monitoring and forecasting efforts.

The quantization of the forecasting model produces an overreacting behavior that is translated into an accuracy drop from 70 % to 50 % for the 28-days' time horizon. Note that most of the accuracy drop is the result of the introduction of false positive peaks as can be seen in Fig. 6. Even though the presented shallow model achieves better results than

Table 6
Approximated mean power consumption values, in milliwatts, and costs, in euros, of the main elements of the proposed water monitoring system.

Element	Mean Power [mW]	Cost [€]
ESP32	10.0	5.0
Electronics	50.0	15.0
RTC	1.0	5.0
Battery	–	10.0
Total	61.0	35.0

others found in the state-of-the-art, LSTM layers are more affected by standard quantization methods than other architectures. Partial quantization of LSTM layers skipping most critical weights (Rezk et al., 2022) or specific quantization methods for LSTM networks (Wang et al., 2022) could be used to reduce the false positives and observed accuracy drop. Alternatively, a less aggressive quantization could be used to trade-off power and accuracy. For example, from 32-bits floating point to 16-bits floating point, 16-bits integers or mixed data types, combined with low-power accelerators (Rusci et al., 2022). Potentially reducing the gap between the original model and the quantized version. Future proposed works also include the implementation of alternative edge devices like AI-accelerators (Rusci et al., 2022). While these devices are expected to reduce the quantization error, they fall outside the scope of this work, which focuses on providing a cost-effective and easy-to-implement solution framework.

The presented model can correctly estimate the trend of the cyanobacteria to predict the appearance of blooms, as can be observed in Fig. 6. A solution to compensate for false positives of the presented embedded monitoring system can be to introduce expert's supervision or to implement a tandem solution where the original shallow model is triggered whenever a bloom is detected by its embedded version, reducing the computational needs until it is required.

8. Conclusions

In this paper, we have proposed a framework introducing a shallow neural network with state-of-the-art DL techniques for embedded bloom forecasting.

The proposed shallow model achieved an accuracy around 70 % for the four studied time horizons. After its training, we quantized the resulting model for 8-bit integers. This reduced the model size resulting in an inference latency two orders of magnitude lower, allowing the model to be executed in the proposed cost-effective edge computing device. The quantized model is able to forecast phycocyanin peaks with a good performance in shape, magnitude, and timing while only introducing false positives peaks. However, since the predictions and raw measurements are broadcasted through wireless communication, they are accessible to any user without additional infrastructure. This approach allows the operator to verify on-site these peak events, counteracting the flaws of the model. Future works will include studying quantization techniques with a lower impact over the forecasting accuracy of the shallow model like QAT, which is not yet supported for recurrent neural networks in the main ML frameworks at the moment of developing this work. Additionally, defining preventive and mitigation measures using model forecasts, and assessing the results obtained in the paper in a real-world scenario by deploying the proposed real-time forecasting add-on for monitoring buoys. Our edge-based system represents a paradigm shift in bloom forecasting, offering a robust, real-time solution that is both cost-effective and adaptable to diverse aquatic environments.

CRediT authorship contribution statement

Juan Sandubete-López: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Raul Fernandez-Fernandez:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **José A. Lopez-Orozco:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization. **José L. Risco-Martín:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

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.

Acknowledgement

This work has been supported by the Research Projects IA-GES-BLOOM-CM (Y2020/TCS-6420) of the Synergic program of the Comunidad Autónoma de Madrid, SMART-BLOOMS (TED2021–130123B-I00) funded by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR, INSERTION (PID2021127648OB-C33) of the Knowledge Generation Projects program of the Spanish Ministry of Science and Innovation, and the MSCA-ITNH2020 Project LasIonDef (GA n.956387).

Data availability

The authors have not permission to share the data, but the code will be open sourced.

References

Aguilera, A., Almanza, V., Haakonsson, S., Palacio, H., Rodas, G.A.B., Barros, M.U., Capelo-Neto, J., Urrutia, R., Aubriot, L., Bonilla, S., 2023. Cyanobacterial bloom monitoring and assessment in latin america. Harmful. Algae 125, 102429. https://doi.org/10.1016/j.hal.2023.102429.

Ahn, C.Y., Oh, H.M., Park, Y.S., 2011. Evaluation of environmental factors on cyanobacterial bloom in eutrophic reservoir using artificial neural networks 1. J. Phycol. 47, 495–504. https://doi.org/10.1111/j.1529-8817.2011.00990.x.

Ahn, J.M., Kim, J., Kim, H., Kim, K., 2023. Harmful cyanobacterial blooms forecasting based on improved cnn-transformer and temporal fusion transformer. Environ. Technol. Innov. 32, 103314. https://doi.org/10.1016/j.eti.2023.103314.

Barrowman, P., Adams, H., Southard, M., Hoppe-Jones, C., Weinrich, L., McNaught, K.J., Clay, K., 2024. Flow imaging microscopy for harmful algal bloom monitoring. J. AWWA 116 (3), 36–48. https://doi.org/10.1002/awwa.2247.

Bengio, Y., Simard, P., Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEe Trans. Neural Netw. 5, 157–166.

Cao, H., Han, L., Li, L., 2022. A deep learning method for cyanobacterial harmful algae blooms prediction in taihu lake, china. Harmful. Algae 113, 102189. https://doi.org/10.1016/j.hal.2022.102189.

ChatGPT, 2022. Openai's gpt-3.5 model. https://openai.com/gpt-3.

Chen, Q., Guan, T., Yun, L., Li, R., Recknagel, F., 2015. Online forecasting chlorophyll a concentrations by an auto-regressive integrated moving average model: feasibilities and potentials. Harmful. Algae 43, 58–65. https://doi.org/10.1016/j.hal.2015.01.002.

Chimmula, V.K.R., Zhang, L., 2020. Time series forecasting of covid-19 transmission in canada using lstm networks. Chaos Solit. Fractals 135, 109864. https://doi.org/10.1016/j.chaos.2020.109864.

Cho, H., Park, H., 2019. Merged-lstm and multistep prediction of daily chlorophyll-a concentration for algal bloom forecast. In: IOP Conference Series: Earth and Environmental Science. IOP Publishing, 012020. https://doi.org/10.1088/1755-1315/351/1/012020.

Chorus, I., Welker, M., 2021. Exposure to cyanotoxins: understanding it and short-term interventions to prevent it. Toxic Cyanobacteria in Water. CRC Press, pp. 295–400.

Cruz, R.C., Reis Costa, P., Vinga, S., Krippahl, L., Lopes, M.B., 2021. A review of recent machine learning advances for forecasting harmful algal blooms and shellfish contamination. J. Mar. Sci. Eng. 9, 283. https://doi.org/10.3390/jmse9030283.

Dignum, M., Matthijs, H., Pel, R., Laanbroek, H., Mur, L., 2005. Nutrient limitation of freshwater cyanobacteria. Harmful Cyanobact. 65–86.

Fernandez-Fernandez, R., Victores, J.G., Balaguer, C., 2023. Deep robot sketching: an application of deep q-learning networks for human-like sketching. Cogn. Syst. Res. 81, 57–63. https://doi.org/10.1016/j.cogsys.2023.05.004.

Fournier, C., Fernandez-Fernandez, R., Cirés, S., López-Orozco, J.A., Besada-Portas, E., Quesada, A., 2024. Lstm networks provide efficient cyanobacterial blooms forecasting even with incomplete spatio-temporal data. Water. Res. 267, 122553. https://doi.org/10.1016/j.watres.2024.122553.

Gatys, L.A., Ecker, A.S., Bethge, M., 2016. Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2414–2423.

Grandvalet, Y., Canu, S., Boucheron, S., 1997. Noise injection: theoretical prospects. Neural Comput. 9, 1093–1108. https://doi.org/10.1162/neco.1997.9.5.1093.

Hajji, M., Benhala, B., Hamdi, I., 2024. Survey of gradient descent variants and evaluation criteria. In: 2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), pp. 01–07. https://doi.org/10.1109/IRASET60544.2024.10549678.

Han, S., Mao, H., Dally, W.J., 2015. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149.
Hewamalage, H., Bergmeir, C., Bandara, K., 2021. Recurrent neural networks for time series forecasting: current status and future directions. Int. J. Forecast. 37, 388–427. https://doi.org/10.1016/j.ijforecast.2020.06.008.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780.
Huisman, J., Codd, G.A., Paerl, H.W., Ibelings, B.W., Verspagen, J.M., Visser, P.M., 2018. Cyanobacterial blooms. Nat. Rev. Microbiol. 16, 471–483.
Huisman, J., Hulot, F.D., 2005. Population dynamics of harmful cyanobacteria. Factors affecting species composition. Harmful Cyanobact., 143–176doi:10.1007/1-4020-3022-3_7.
Ibelings, B.W., Fastner, J., Bormans, M., Visser, P.M., 2016. Cyanobacterial blooms. Ecology, prevention, mitigation and control: editorial to a cyanocost special issue. Aquat. Ecol. 50, 327–331.
Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pmlr, pp. 448–456.
Iwana, B.K., Uchida, S., 2021. An empirical survey of data augmentation for time series classification with neural networks. PLoS. One 16, e0254841. https://doi.org/10.1371/journal.pone.0254841.
Izydorczyk, K., Carpentier, C., Mrówczyński, J., Wagenvoort, A., Jurczak, T., Tarczyńska, M., 2009. Establishment of an alert level framework for cyanobacteria in drinking water resources by using the algae online analyser for monitoring cyanobacterial chlorophyll a. Water. Res. 43, 989–996. https://doi.org/10.1016/j.watres.2008.11.048.
Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D., 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2704–2713. https://doi.org/10.1109/CVPR.2018.00286.
Joshi, A., Agarwal, S., Kanungo, D.P., Panigrahi, R.K., 2024. Integration of Edge–AI into IoT–Cloud architecture for landslide monitoring and prediction. IEEE Trans. Ind. Inform. 20 (3), 4246–4258. https://doi.org/10.1109/TII.2023.3319671.
Kang, Z., Yang, B., Lai, J., Ning, Y., Zhong, Q., Lu, D., Liao, R., Wang, P., Dan, S.F., She, Z., Jia, Z., Lao, Y., Li, N., 2020. Phaeocystis globosa bloom monitoring: based on p. globosa induced seawater viscosity modification adjacent to a nuclear power plant in qinzhou bay, china. J. Ocean. Univ. China 19, 1207–1220. https://doi.org/10.1007/s11802-020-4481-6.
Kingma, D.P., Ba, J., 2015. Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations (ICLR).
Lim, B., Zohren, S., 2021. Time-series forecasting with deep learning: a survey. Philos. Trans. R. Soc. A 379, 20200209.
Liu, M., He, J., Huang, Y., Tang, T., Hu, J., Xiao, X., 2022. Algal bloom forecasting with time-frequency analysis: a hybrid deep learning approach. Water. Res. 219, 118591. https://doi.org/10.1016/j.watres.2022.118591.
Liu, Y., Wang, H., Feng, W., Huang, H., 2021. Short term real-time rolling forecast of urban river water levels based on lstm: a case study in fuzhou city, china. Int. J. Env. Res. Public Health 18. https://doi.org/10.3390/ijerph18179287.
Martín-Suazo, S., Morón-López, J., Vakaruk, S., Karamchandani, A., Aguilar, J.A.P., Mozo, A., Gómez-Canaval, S., Vinyals, M., Ortiz, J.M., 2024. Deep learning methods for multi-horizon long-term forecasting of harmful algal blooms. Knowl. Based. Syst. 301, 112279. https://doi.org/10.1016/j.knosys.2024.112279.
Masini, R.P., Medeiros, M.C., Mendes, E.F., 2023. Machine learning advances for time series forecasting. J. Econ. Surv. 37, 76–111. https://doi.org/10.1111/joes.12429.
Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on International Conference on Machine Learning. Omnipress, pp. 807–814. https://doi.org/10.5555/3104322.3104425.
Nielsen, M.A., 2015. Neural Networks and Deep Learning, 25. Determination press, San Francisco, CA, USA.
Pescheck, M., Schweizer, A., Bláha, L., 2022. Innovative electrochemical biosensor for toxicological investigations on algae and cyanobacteria. Bioelectrochemistry 143, 107926. https://doi.org/10.1016/j.bioelechem.2021.107926.
Ray, P.P., 2022. A review on tinyml: state-of-the-art and prospects. J. King Saud Univ. - Comput. Inf. Sci. 34, 1595–1623. https://doi.org/10.1016/j.jksuci.2021.11.019.
Rezk, N.M., Nordström, T., Ul-Abdin, Z., 2022. Shrink and eliminate: a study of post-training quantization and repeated operations elimination in RNN models. Information 13 (4), 176. https://doi.org/10.3390/info13040176.
Ribeiro, R., Torgo, L., 2008. A comparative study on predicting algae blooms in Douro river. Port., Ecol. Model. 212, 86–91.
Richardson, T.L., Lawrenz, E., Pinckney, J.L., Guajardo, R.C., Walker, E.A., Paerl, H.W., MacIntyre, H.L., 2010. Spectral fluorometric characterization of phytoplankton community composition using the algae online analyser®. Water. Res. 44, 2461–2472. https://doi.org/10.1016/j.watres.2010.01.012.
Ruiz-Villarreal, M., Sourisseau, M., Anderson, P., Cusack, C., Neira, P., Silke, J., Rodriguez, F., Ben-Gigirey, B., Whyte, C., Giraudeau-Potel, S., Quemener, L., Arthur, G., Davidson, K., 2022. Novel methodologies for providing in situ data to hab early warning systems in the European atlantic area: the primrose experience. Front. Mar. Sci. 9. https://doi.org/10.3389/fmars.2022.791329.
Rusci, M., Fariselli, M., Capotondi, A., Benini, L., 2020. Leveraging automated mixed-low-precision quantization for tiny edge microcontrollers. Springer Int. Publ. 296–308. https://doi.org/10.1007/978-3-030-66770-2_22.
Rusci, M., Fariselli, M., Croome, M., Paci, F., Flamand, E., 2022. Accelerating rnn-based speech enhancement on a multi-core mcu with mixed fp16-int8 post-training quantization. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Nature Switzerland, pp. 606–617.
Sandonnini, J., Del Pilar Ruso, Y., Cortés Melendreras, E., Barberá, C., Hendriks, I.E., Kersting, D.K., Giménez Casalduero, F., 2021. The emergent fouling population after severe eutrophication in the Mar Menor coastal lagoon. Reg. Stud. Mar. Sci. 44, 101720. https://doi.org/10.1016/j.rsma.2021.101720.
Smith, V.H., 2003. Eutrophication of freshwater and coastal marine ecosystems a global problem. Environ. Sci. Pollut. Res. 10, 126–139. https://doi.org/10.1065/espr2002.12.142.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.
Sun, C., Shrivastava, A., Singh, S., Gupta, A., 2017. Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE international conference on computer vision, pp. 843–852.
Turley, B.D., Karnauskas, M., Campbell, M.D., Hanisko, D.S., Kelble, C.R., 2022. Relationships between blooms of karenia brevis and hypoxia across the West Florida Shelf. Harmful. Algae 114, 102223. https://doi.org/10.1016/j.hal.2022.102223.
Wang, Y., Ma, Z., Yang, Z., 2022. Sequential characteristics based operators disassembly quantization method for LSTM layers. Appl. Sci. 12 (24), 12744. https://doi.org/10.3390/app122412744.
Wu, H., Judd, P., Zhang, X., Isaev, M., Micikevicius, P., 2020. Integer quantization for deep learning inference: principles and empirical evaluation. arXiv preprint arXiv:2004.09602.
Xiao, X., Peng, Y., Zhang, W., Yang, X., Zhang, Z., Ren, B., Zhu, G., Zhou, S., 2024. Current status and prospects of algal bloom early warning technologies: a review. J. Env. Manage 349, 119510. https://doi.org/10.1016/j.jenvman.2023.119510.
Yajima, H., Derot, J., 2018. Application of the random forest model for chlorophyll-a forecasts in fresh and brackish water bodies in japan, using multivariate long-term databases. J. Hydroinformat. 20, 206–220. https://doi.org/10.2166/hydro.2017.010.
You, L., Tong, X., Te, S.H., Tran, N.H., bte Sukarji, N.H., He, Y., Gin, K.Y.H., 2022. Multi-class secondary metabolites in cyanobacterial blooms from a tropical water body: distribution patterns and real-time prediction. Water. Res. 212, 118129. https://doi.org/10.1016/j.watres.2022.118129.
Yuan, A., Wang, B., Li, J., Lee, J.H.W., 2023. A low-cost edge AI-chip-based system for real-time algae species classification and HAB prediction. Water. Res. 233, 119727. https://doi.org/10.1016/j.watres.2023.119727.
Zahir, M., Su, Y., Shahzad, M.I., Ayub, G., Rahman, S.U., Ijaz, J., 2024. A review on monitoring, forecasting, and early warning of harmful algal bloom. Aquaculture 593, 741351. https://doi.org/10.1016/j.aquaculture.2024.741351.
Zheng, L., Wang, H., Liu, C., Zhang, S., Ding, A., Xie, E., Li, J., Wang, S., 2021. Prediction of harmful algal blooms in large water bodies using the combined efdc and lstm models. J. Env. Manage 295, 113060. https://doi.org/10.1016/j.jenvman.2021.113060.