water

MDPI

Article

IWQP4Net: An Efficient Convolution Neural Network for Irrigation Water Quality Prediction

Ibrahim Al-Shourbaji * **and Salahaldeen Duraibi**

Department of Computer and Network Engineering, Jazan University, Jazan 45142, Saudi Arabia; sduraibi@jazanu.edu.sa
* Correspondence: alshourbajiibrahim@gmail.com

Abstract: With the increasing worldwide population and the requirement for efficient approaches to farm care and irrigation, the demand for water is constantly rising, and water resources are becoming scarce. This has led to the development of smart water management systems that aim to improve the efficiency of water management. This paper pioneers an effective Irrigation Water Quality Prediction (IWQP) model using a convolution neural architecture that can be trained on any general computing device. The developed IWQP4Net is assessed using several evaluation measurements and compared to the Logistic Regression (LR), Support Vector regression (SVR), and k-Nearest Neighbor (kNN) models. The results show that the developed IWQP4Net achieved a promising outcome and better performance than the other comparative models.

Keywords: irrigation system; water quality; deep learning; convolutional neural network prediction

1. Introduction

The United Nations (UN) has asserted that access to safe domestic potable water is an essential human right and a step towards improving people's living standards. An accessible, adequate, clean, and safe drinking water supply is one of the main aspirations of the Millennium Development Goals (UN-MDGs), and it is one of the core aims of the Sustainable Development Goals (SDGs) [1]. The UN-MDGs concluded that water sustains life, but safe and clean drinking water is a hallmark of civilization. In order to ensure that consumers receive the required quality of water supplied from treatment plants, there is a need to deter, detect and minimize any threats to the water supply such as accidents, backflow, vandalism, hazardous material releases, and terrorism. Natural disasters such as waterborne disease outbreaks, bushfires, and floods can also affect the quality of water supplied. The supply of safe potable water to consumers implies that there must be proper security setups and safeguards in terms of how rapidly to react to disasters and when they occur [2].

A significant amount of freshwater is used in farming and agriculture, especially for irrigation. Due to a lack of affordable clever irrigation technologies, developing nations use more water than industrialized nations to produce the same output. For instance, India has only 4% of the world's freshwater resources to service 17% of the global population, but some of its key agricultural products require two to four times more water than those of other nations such as China and the USA [3]. Therefore, there is an urgent need to develop smart methods and systems based on sophisticated technologies for the efficient use of fresh water.

Many studies have used Deep Learning (DL) approaches for irrigation water systems [4–7]. The Internet of Things (IoT) was used by the authors of [8] to collect data from the field on a variety of variables, including soil moisture, soil temperature, weather, and environmental conditions. Their strategy optimizes the nee and thereby uses less energy with the aid of different information. The collected data from the IoT devices were saved in the cloud, and they then employed ML models for precise irrigation systems.

check for updates

Citation: Al-Shourbaji, I.; Duraibi, S. IWQP4Net: An Efficient Convolution Neural Network for Irrigation Water Quality Prediction. *Water* **2023**, *15*, 1657. https://doi.org/10.3390/w15091657

Academic Editors: Padam Jee Omar, Ganesh Ji Omar and Pankaj Kumar Gupta

Received: 19 March 2023
Revised: 17 April 2023
Accepted: 20 April 2023
Published: 24 April 2023

In [9], an intelligent irrigation system for precision agriculture based on DL neural networks is explored. The developed method can maintain its functionality better over time and in any weather conditions. It uses an extended short-term memory network (LSTM) to forecast the irrigation schedule, the spatial spread of water needed to irrigate the arable land, and the volumetric soil moisture content for the following day. In [10], the authors used IoT sensors to collect soil moisture, temperature, humidity, and time data, and then they used that data as inputs to LR, Random Forest (RF), SVR, and Convolutional Neural Network (CNN) techniques to classify the produced data from the IoT. Based on the results, the RF model performed the best compared to the other techniques.

In the study of [11], the regression analysis method showed better results than the correlation for monitoring water quality and predicting trends in water quality variation. In [12], an automation approach for the farm irrigation system was reported based on IoT and ML. The authors established a distributed wireless sensor network (WSN), where various sensors cover each region of the farm, and then these data were transmitted to a server. They then used stored data were used to predict irrigation patterns using ML methods.

In order to estimate the infiltrated water in the furrow irrigation system, the authors of [13] applied the SVR, Group Method of Data Handling (GMDH), Adaptive Neuro-Fuzzy Inference System (ANFIS), Artificial Neural Network (ANN), and Multivariate Linear Regression (MLR) models. The results showed that the Firefly Algorithm (FA) has a high ability to boost the accuracy of the implemented models. The authors in [14] present a machine learning (ML) algorithm-based smart system to anticipate irrigation needs by sensing ground parameters such as soil moisture, temperature, humidity, and water level. The authors in [15] discussed the recent contributions of both DL and IoT for precision agriculture. They also proposed a boot-strapping method of transfer learning, where fine-tuned VGG16 is improved by newly built fully connected layers. The results showed that their improvement method accomplished better accuracy than the other models used in their work.

In [16], a stacking model of ML combined with feature selection methods was investigated to accurately calculate the required amount of water for a plant. The relevance of the features was determined by using RF, state-of-the-art feature elimination, and SelectKBest. An ensemble of CART, gradient boost regression, and XGBoost regressors was suggested based on the best subset of characteristics. Several crops, such as tomatoes, grapes, and lemons, were used to train and test the models, which incorporated information on the weather, soil, irrigation, and crops. The feature selection results showed the significance of evapotranspiration, depletion, and shortfall levels in maximizing the model's accuracy.

In [17], the authors investigated several ML-based methods and their performance to calculate Water Quality Index (WQI) using a dataset collected from the Kim Hai irrigation system. They applied several feature selection methods to pick key parameters fed to the ML models. Results confirmed that a combination of feature selection along with ML has been an effective choice for calculating the WQI. In [18], Long Short-Term Memory LSTM, CNN, and hybrid CNN-LSTM models, were examined in order to forecast water quality. The results showed that the hybrid model had a high ability to capture low and high levels of water quality variables compared to LSTM and CNN alone. As a result, improved water quality prediction still requires more sophisticated prediction algorithms to deliver an effective approach with high performance. The main contributions of this paper can be summarized as follows:

1. Introduced an effective Irrigation Water Quality Prediction (IWQP) approach based on an efficient CNN architecture which comprises two convolution layers, a max-pooling layer, a dropout layer, and two dense layers;
2. Examined and compared the performance of IWQP4Net with the LR, SVR, and kNN models using different evaluation measurements and visualizations methods;
3. Demonstrated the efficiency of the developed IWQP4Net for IWQP and its superiority over the other competitive models.

The remaining parts of this paper consist of four sections. Section 2 presents an overview of the dataset used in this work and the developed IWQP approach. Section 3 provides the evaluation measurements used to evaluate the introduced IWQP model. Section 4 discusses the experimental results of this work. Finally, Section 5 presents the conclusion of this paper.

2. Methods and Material

2.1. Dataset

On the basis of the input data, which is the historical data of other water-measurement indices, the spatiotemporal water quality is to be forecast in terms of the "power of hydrogen (pH)" value for the following day. The United States Geological Survey publishes this dataset [19]. Depending on the water system they are a part of, such as the water system of Atlanta, the Savannah River's watershed, and others, high-level prior spatial information is given. It comprises training data collected for 423 days (1 December 2014 to 28 January 2016), and testing data collected for 282 days (25 March 2017 to 1 January 2018). The input data comprises daily samples taken from 37 water stations and that were used to calculate the pH levels in Georgia, USA. Due to the complexity of the water system, the precise connections between each location through water streams, or spatial connectivity, are unknown. Eleven standard parameters, including temperature, specific conductance, and the volume of dissolved oxygen, are used to predict the pH value for each water station. The input data at any time instant is spatially arranged as a matrix of dimension (37×11), where rows correspond to different water stations and columns correspond to different features affecting the pH value. Hence, the training input data has a dimension of ($423 \times 37 \times 11$), and a concatenation of 423 days of spatial matrices. Similarly, the test input data has a dimension of ($282 \times 37 \times 11$). The output comprises the pH value for 37 water stations for each day. Hence, the training and test output data dimensions are (423×37) and (282×37), respectively. A statistical summary of input and output training and testing data is presented in Table 1. It can be assumed that both data are generated from the same distribution and, hence, can be used to train and evaluate the CNN for IWQP.

Table 1. Statistical summary of training and testing input data and corresponding pH values.

	Measure	Feat-1	Feat-2	Feat-3	Feat-4	Feat-5	Feat-6	Feat-7	Feat-8	Feat-9	Feat-10	Feat-11	pH
Training	Mean	0.0652	0.8894	0.0290	0.0432	0.5678	0.8586	0.6049	0.5783	0.5571	0.5368	0.5532	0.6644
	S.D.	0.1614	0.0348	0.1208	0.1333	0.1205	0.0311	0.1468	0.1718	0.2040	0.2123	0.1901	0.0293
	Min	0.0005	0.5769	0.0003	0.0005	0.1250	0.7195	0.0758	0.0315	0.0594	0.0256	0.0901	0.5741
	Q1	0.0019	0.8718	0.0016	0.0018	0.4868	0.8415	0.5152	0.4803	0.3938	0.3654	0.3983	0.6481
	Q2	0.0026	0.8974	0.0022	0.0024	0.5658	0.8537	0.6061	0.5906	0.5313	0.5096	0.5320	0.6667
	Q3	0.0055	0.9103	0.0035	0.0048	0.6579	0.8780	0.7121	0.7087	0.7406	0.7244	0.7238	0.6759
	Max	1.0000	0.9872	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9630
Testing	Mean	0.0813	0.8843	0.0340	0.0529	0.5256	0.8552	0.5556	0.5259	0.6352	0.6188	0.6244	0.6622
	S.D.	0.1780	0.0360	0.1243	0.1415	0.1138	0.0312	0.1405	0.1695	0.1852	0.1929	0.1724	0.0294
	Min	0.0007	0.4103	0.0003	0.0007	0.1184	0.7317	0.0682	0.0315	0.0625	0.0224	0.1105	0.5741
	Q1	0.0019	0.8590	0.0015	0.0018	0.4671	0.8415	0.5000	0.4646	0.5281	0.5032	0.5291	0.6481
	Q2	0.0026	0.8846	0.0022	0.0024	0.5263	0.8537	0.5682	0.5512	0.6750	0.6571	0.6613	0.6574
	Q3	0.0056	0.9103	0.0040	0.0050	0.5921	0.8780	0.6288	0.6142	0.7656	0.7532	0.7500	0.6759
	Max	1.0000	1.0000	0.8566	0.8633	0.9276	0.9878	0.9848	0.9606	1.0000	0.9936	0.9913	1.0000

2.2. Proposed IWQP Approach

DL methods have attracted significant research interest due to their response capability to many data forms in diverse contexts, such as prediction and classification issues. They have been successfully applied to smart homes [20], image segmentation [21], self-driving vehicles [22], agriculture [23], and to many others. These techniques teach a device to analyse inputs using several layers, similar to human brain processing, to aid in data classification and prediction, simulating the brain's capacity to process information and reach the right judgments [24]. In this work, the water quality dataset requires a quality value to be predicted for each water station. This indicates that the given problem is multivariate regression, with 37 outputs to be predicted simultaneously. Figure 1 shows the overall system flow used in this work to predict the water quality in terms of pH value in the irrigation system framework. The input data is already divided into training and test subsets. We use only the training subset for model generation and validation, while the testing subset is used to evaluate the generalized performance of the model using three different errors.

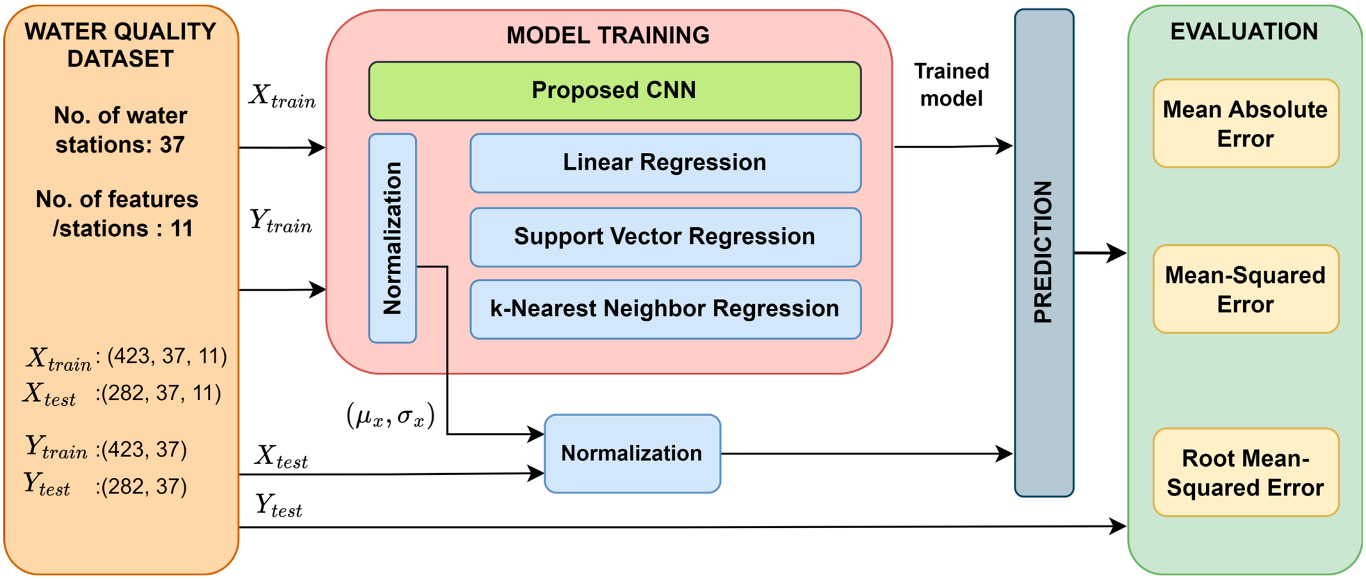

Figure 1. Overall system flow of the IWQP approach.

The developed architecture of CNN for irrigation water quality prediction (IWQP4Net) is shown in Figure 2. It comprises two convolution layers, a max-pooling layer, a dropout layer, and two fully-connected (dense) layers. The input data is arranged in a 3D matrix as: (no. of training examples, no. of water stations, no. of features per station). The training and test data dimensions are ($423 \times 37 \times 11$) and ($282 \times 37 \times 11$), respectively. Hence, 2D convolution and 2D max-pooling layers are used to analyze the data [25]. The initial convolution layer uses 20 kernel functions of (3×3) dimensions to extract the coarse features. For the ith layer, the output $y^{[i]}(j,\ k)$ at jth row and kth column is calculated by convolving the kernel $w^{[i]}$ with the output of the previous layer $\{y^{[i-1]}(j,\ k)$ s.t. $j \in [0,\ J-1]$ & $k \in [0,\ K-1]\}$ as,

$$y^{[i]}(j,k) = \sum_{a=0}^{2}\sum_{b=0}^{2} w^{[i]}(j,k)\ y^{[i-1]}(j+a,\ k+b) \tag{1}$$

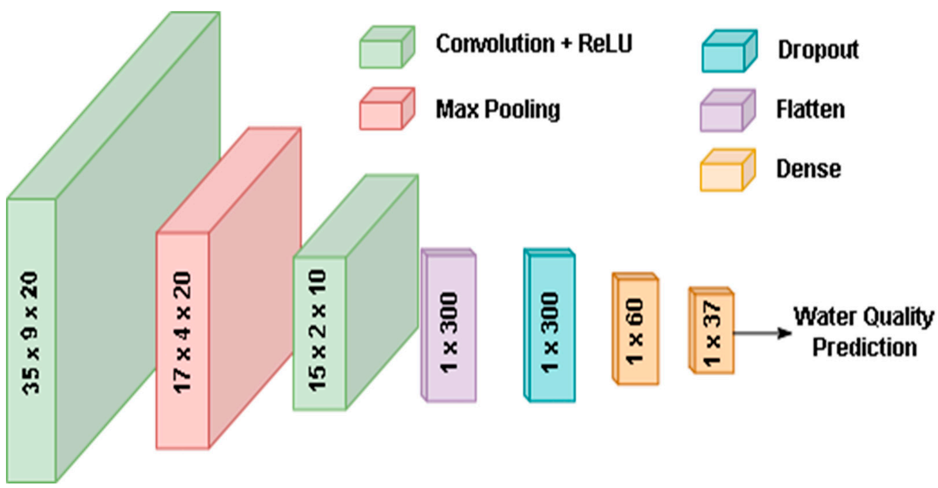

Figure 2. The architecture of the developed IWQP4Net.

The output dimension of the first convolution layer is (35 × 9 × 20), corresponding to a reduction of 2 in the first two dimensions by kernel convolution, and the third dimension is the same as the number of individual kernel functions. The zero-padding is not used to eliminate redundant data and bottleneck sooner in the network, forcing only important features to pass further. The dimension of the first layer's output is further reduced by a max-pooing layer that uses a (2 × 2) window function. With similar assumptions as above, the output of any max-pooling layer can be calculated as,

$$y^{[i]}(j,k) = \max\left(y^{[i-1]}(j,\ k),\ y^{[i-1]}(j+1,\ k),\ y^{[i-1]}(j,\ k+1),\ y^{[i-1]}(j+1,\ k+1)\right) \quad (2)$$

In simple terms, only the most significant (highest-valued) convolutions are retained, while others are discarded. The output dimension of max-pooled convolution is (17 × 4 × 20), compressing the first two dimensions by the respective kernel size while the number of filters dimension is retained. To generate more complex features that capture the finer details, another convolution layer comprising ten kernel functions, each of (3 × 3) dimension, is added. To maintain our objective, zero-padding was not allowed in this layer, similar to the previous convolution layer. Hence, the output dimension reduces to (15 × 2 × 10). This ends the feature extractor part of the IWQP4Net. The details of layers, output dimension, and trainable parameters are provided in Table 2.

Table 2. Developed IWQP4Net model's parameters summary.

Layer Type	Kernel Parameters	Output Shape	Trainable Parameters
Input	-	(37, 11)	-
Convolution 2D	(3, 3, 20), ReLU	(35, 9, 20)	200
Max. Pooling 2D	(2, 2)	(17, 4, 20)	0
Convolution 2D	(3, 3, 10), ReLU	(15, 2, 10)	1810
Flatten	(1, 300)	(300)	0
Dropout	0.4	(300)	0
Dense	60, ReLU	(60)	18,060
Dense	37, ReLU	(37)	2257
Total trainable parameters			**22,327**

The 3D output of the feature extractor part is rearranged to only one dimension by the flattened layer. It should be noted that the flattened layer, the fourth layer shown in the IWQP4Net architecture, is non-trainable and has no parameters. It can be understood as a cylindrical convolution layer with a unity valued kernel function of size (1 × 1) and the number of filters equal to the total number of coefficients in the previous convolution layer output. The resulting dimension is 300 (15 × 2 × 10). The large number of parameters in the CNNs keeps them at risk of overfitting. In this work, training examples are 423, which a large CNN model can easily memorize. To avoid overfitting and increase generalization [26,27], a dropout layer with 40% of the input being randomly set to 0 is added in the sequential model. This is another example of a non-trainable layer. It is followed by another dense layer with 60 neurons that acts as a reduction layer to filter out important features for the current regression task. Finally, an output layer with the number of neurons equal to the number of water stations is added to get the final water quality prediction. All trainable layers in the IWQP4Net architecture use Rectified Linear Unit (ReLU) activation.

3. Evaluation Measurements

The introduced approach and the other models are evaluated using statistical evaluation criteria as given in Table 3. Also, visualization approaches comprising the loss curve of training and validation, scatter plots of prediction and observed values, and the histogram of prediction error with Gaussian approximation are used to assess the models. For the observed water quality WQ_{obs} and predicted water quality WQ_{pre}, the evaluation measures can be calculated as follows:

Table 3. Evaluation measures of the models.

Measure	Definition	Equation	
RMSE	Root Mean Squared Error	$RMSE = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(WQ_{obs}(m) - WQ_{pre}(m))^2}$	(3)
MAPE	Mean Absolute Percentage Error	$MAPE \ = \frac{1}{M}\sum_{m=1}^{M}\lvert WQ_{obs}(m) - WQ_{pre}(m)/WQ_{obs}(m)\rvert$	(4)
MAE	Mean Absolute Error	$MAE = \frac{1}{M}\sum_{m=1}^{M}\lvert WQ_{obs}(m) - WQ_{pre}(m)\rvert$	(5)

4. Experimental Results and Discussion

The open-source TensorFlow library in Python is used to implement all of the models, which are then run on an Intel Xenon CPU running at 2.00 GHz with 13 GB of memory and a 12-GB NVIDIA Tesla P100 GPU. Table 4 shows the hyper-parameters used in the modeling stage of the LR, SVR, kNN, and IWQP models. These settings are chosen after experimental research showed that they are the most effective ones for model training.

Table 4. Model's training of the hyper-parameter settings.

Models	Hyper-Parameters and Value
LR	Fit intercept = *True*, Normalization = *True*
SVR	Kernel= *Radial basis Function*, Normalization = *True*, C = 1.0, epsilon = 0.2
kNN	Neighbors = 5, weights = *Uniform*, leaf size = 30, metric = *Minkowski*
IWQP4Net	Loss = *MSE*, optimizer = *Adam*, learning rate = 0.01, batch size = 120, epochs = 80, validation split = 0.2

The IWQP4Net has a large number of trainable parameters, as shown in Table 1. The 423 training examples are available in the dataset, which is very small compared to the trainable parameters of the IWQP4Net. This puts the network at risk of overfitting. During training, the optimum hyper-parameter set for the network was guided by the loss curves for training and validation. The learning rate and batch size are searched over a logarithmic range, while epochs are varied linearly. For large learning rates, the loss function is noisy; hence, the CNN is unreliable due to the high variance problem. This variance can be reduced up to a level by increasing the batch size, but the predicted MAPE was high. After reducing the learning rates to very small values, the optimum solution was obtained after several hundreds of epochs. Figure 3 shows the training and validation loss for the developed IWQP4Net's optimum hyper-parameters, as mentioned in Table 4. It can be seen that both loss values decrease rapidly at the start, and no overfitting has occurred.

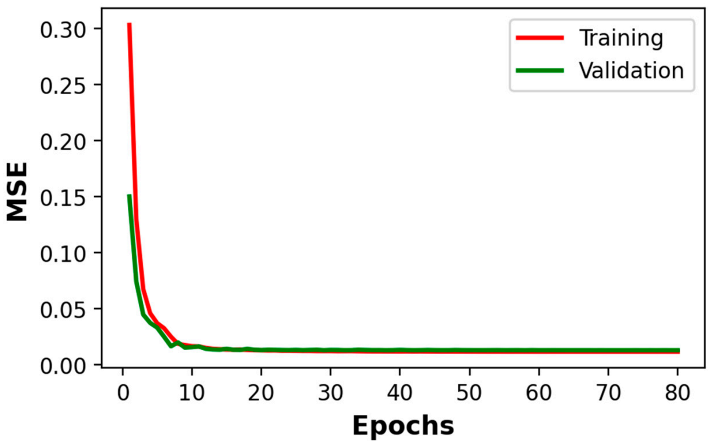

Figure 3. Mean squared error of the IWQP4Net model with optimum hyper-parameters for the training and validation data of water quality prediction.

The average training and testing performance of all competing models for 10 iterations are tabulated in Table 5, using the three types of evaluation measures provided in Section 3. The training errors for LR are the lowest, indicating the best training performance. However, the testing performance of the LR model is the worst. This indicates that the LR model suffers from an overfitting problem. The developed IWQP4Net approach shows the second-highest training performance, followed by kNN and SVR. All three errors of the developed IWQP4Net are lower during testing than the three competing models, which indicates its higher capability than the others for water quality prediction.

Table 5. Comparative analysis of different prediction models using error measures.

Model	Data Subset	RMSE	MAPE	MAE
LR	Training	0.0008	0.0009	0.0006
	Testing	0.0549	0.0855	0.0406
SVR	Training	0.0122	0.0251	0.0097
	Testing	0.0100	0.0211	0.0085
kNN	Training	0.0052	0.0085	0.0036
	Testing	0.0108	0.0212	0.0090
IWQP4Net	Training	0.0048	0.0078	0.0034
	Testing	0.0080	0.0195	0.0068

Further investigation of the performance of the introduced approach is also carried out by visualizing the scatter plots. Figure 4 shows scatter plots with the observed pH predicted on the horizontal axis and the predicted pH on the vertical axis of all four models for training and testing data. Each scatter plot is superimposed with a regression line with the regression equation mentioned at the top. As shown in Table 5, LR has the smallest training MAPE and the testing MAPE is the worst. A similar visualization can be seen in Figure 4.

The scatter plot for LR training shows the best performance with a minimal deviation for the regression line, but testing scatter plots show a very high deviation from the regression line. Scatter plots for the IWQP4Net model show a good match between the observed and predicted water quality. The SVR and kNN models show slightly higher deviation from the regression line than the developed model.

The prediction error analysis is examined to understand the behavior of all four prediction models. Figure 5 shows the histograms (bin width = 0.005) of prediction error ($WQ_{pre} - WQ_{obs}$) of all four models for training and testing data. Kernel density (KDE) plots superimpose all histograms with Gaussian kernel and bandwidth calculated using Scott's method [28]. As stated above, an impulsive training error and an almost flat (very high standard deviation) testing error characterizes the overfitted LR model. The training histogram and KDE plots of IWQP4Net provide a high concentration with lower prediction errors. Similar behavior can also be seen in the testing data with a slight bias toward the value for predicted water quality. The kNN model also shows a concentration of training and testing errors toward lower values with a slightly higher standard deviation than the developed method. The SVR model shows a significantly large variation in training and testing errors.

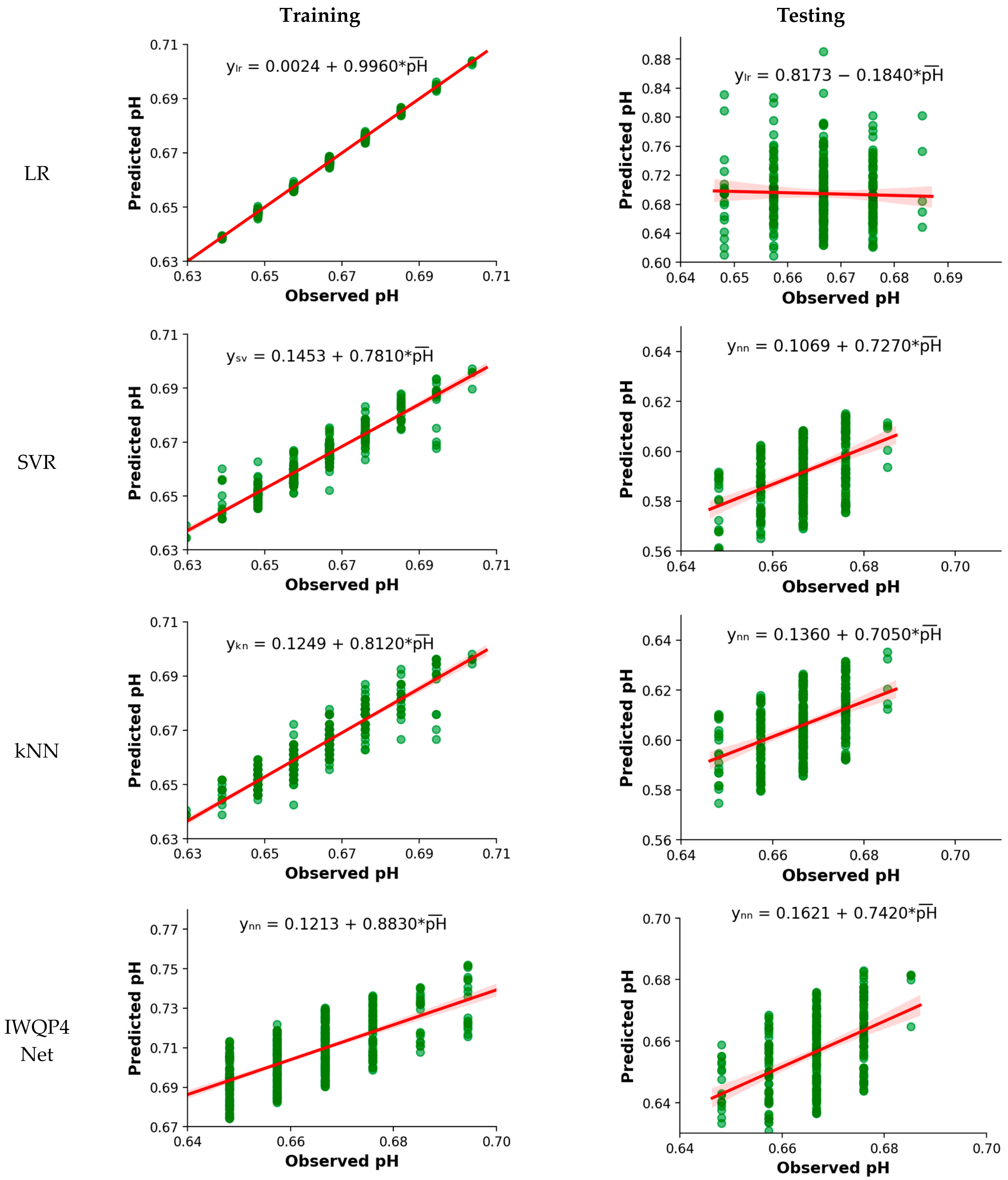

Figure 4. Scatter plots of observed and predicted pH (marked by green dot) with regression (red solid) for training and testing data.

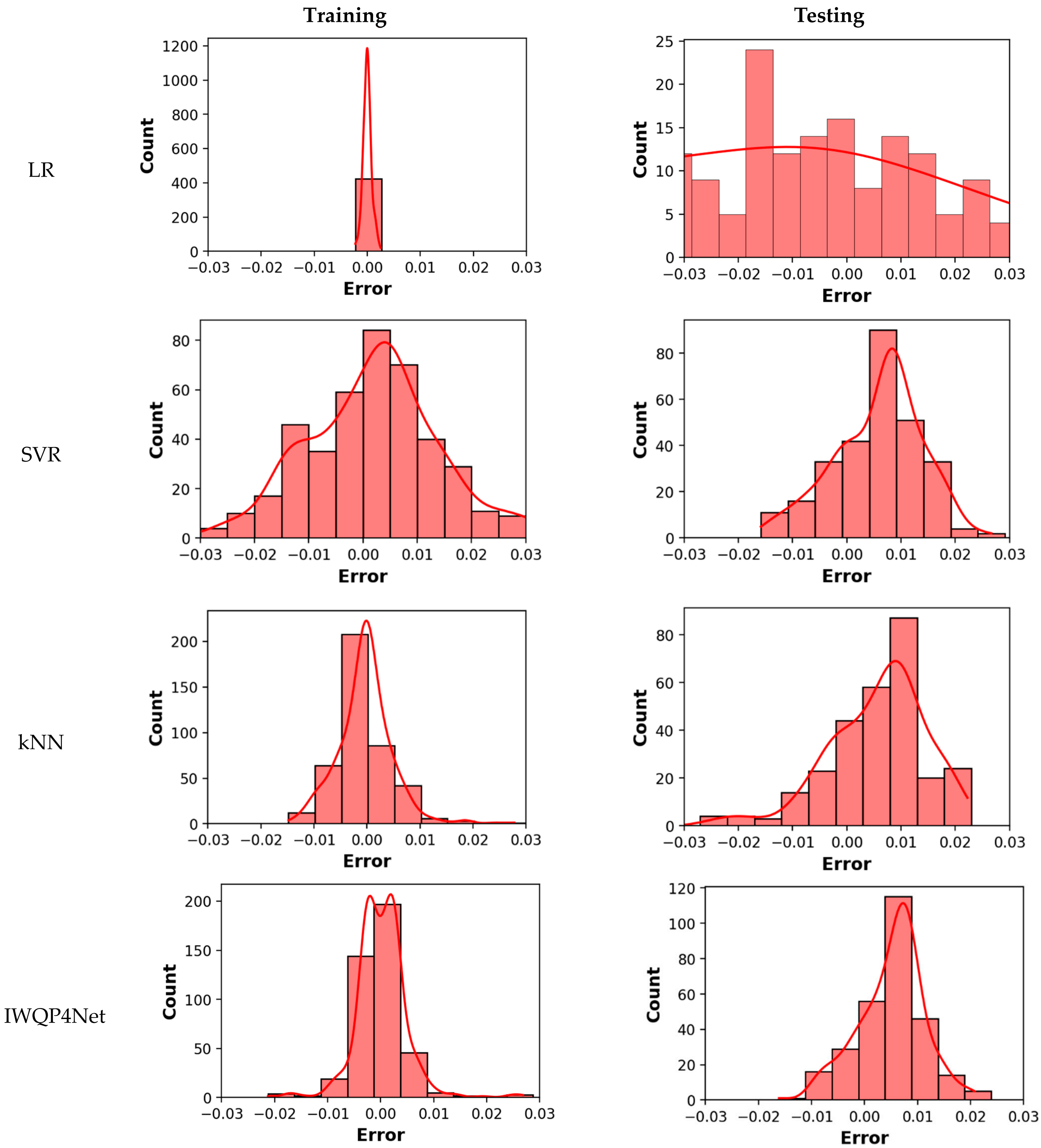

Figure 5. Histogram of the observed error and predicted pH with kernel density approximation (red solid) for training and testing data.

5. Conclusions and Future Works

The recent years have seen a rise in the interest in precision agriculture due to the increasing demands for food and water from a population that is expanding worldwide. To satisfy these demands, farmers thus need access to water and fertile land. In this paper, an IWQP approach is developed by using an efficient CNN structure. The IWQP4Net consists of two convolution layers, a max-pooling layer, a dropout layer, and two dense layers. Some statistical measures, including RMSE, MSE, and MAE, are utilized to assess

IWQP4Net, and its performance is compared with the LR, SVR, and kNN models. The results showed that the developed IWQP4Net approach has the lowest RMSE, MSE, and MAE values compared to the other models. In the future, the proposed IWQP approach can be applied to a large-scale IoT-based online monitoring system to predict the water quality based on the real-time data fed from the IoT system. Another possible avenue is to apply metaheuristic methods as a feature selection along with the developed IWQP4Net approach in the application of agriculture, because these optimization algorithms have shown great potential in other domains.

Author Contributions: Methodology, I.A.-S. and S.D.; software I.A.-S., resources, I.A.-S.; data curation, I.A.-S. and S.D.; writing—original draft preparation, I.A.-S.; writing—review, I.A.-S. and S.D. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the deanship for research and innovation, and to the Ministry of Education in Saudi Arabia for funding this research work through project number RUP3-1.

Data Availability Statement: The data supporting the findings of this study are publicly available in Kaggle.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Li, P.; Wu, J. Drinking Water Quality and Public Health. *Expo. Health* **2019**, *11*, 73–79. [CrossRef]
2. Fadel, A.A.; Shujaa, M.I. Water Quality Monitoring System Based on IOT Platform. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *928*, 032054. [CrossRef]
3. National Institution for Transforming India (NITI) Aayog. *Raising Agricultural Productivity and Making Farming Remunerative for Farmers*; NITI Aayog, Government of India: New Delhi, India, 2015.
4. Kondaveti, R.; Reddy, A.; Palabtla, S. Smart irrigation system using machine learning and IoT. In Proceedings of the 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, 30–31 March 2019.
5. Goap, A.; Sharma, D.; Shukla, A.K.; Krishna, C.R. An IoT based smart irrigation management system using Machine learning and open source technologies. *Comput. Electron. Agric.* **2018**, *155*, 41–49. [CrossRef]
6. Saraiva, M.; Protas, É.; Salgado, M.; Souza, C., Jr. Automatic mapping of center pivot irrigation systems from satellite images using deep learning. *Remote Sens.* **2020**, *12*, 558. [CrossRef]
7. Chen, H.; Chen, A.; Xu, L.; Xie, H.; Qiao, H.; Lin, Q.; Cai, K. A deep learning CNN architecture applied in smart near-infrared analysis of water pollution for agricultural irrigation resources. *Agric. Water Manag.* **2020**, *240*, 106303. [CrossRef]
8. Vianny, D.M.M.; John, A.; Mohan, S.K.; Sarlan, A.; Adimoolam; Ahmadian, A. Water optimization technique for precision irrigation system using IoT and machine learning. *Sustain. Energy Technol. Assess.* **2022**, *52*, 102307. [CrossRef]
9. Kashyap, P.K.; Kumar, S.; Jaiswal, A.; Prasad, M.; Gandomi, A.H. Towards precision agriculture: IoT-enabled intelligent irrigation systems using deep learning neural network. *IEEE Sens. J.* **2021**, *21*, 17479–17491. [CrossRef]
10. Iorliam, A.; Sylvester, B.U.M.; Aondoakaa, I.S.; Iorliam, I.B.; Shehu, Y. Machine Learning Techniques for the Classification of IoT-Enabled Smart Irrigation Data for Agricultural Purposes. *Gazi Univ. J. Sci. Part A Eng. Innov.* **2022**, *9*, 378–391. [CrossRef]
11. Nayak, S. Assessment of Water Quality of Brahmani River Using Correlation and Regression Analysis. 2020. Available online: https://www.preprints.org/manuscript/202003.0088/v1 (accessed on 18 March 2023).
12. Vij, A.; Vijendra, S.; Jain, A.; Bajaj, S.; Bassi, A.; Sharma, A. IoT and Machine Learning Approaches for Automation of Farm Irrigation System. *Procedia Comput. Sci.* **2020**, *167*, 1250–1257. [CrossRef]
13. Sayari, S.; Mahdavi-Meymand, A.; Zounemat-Kermani, M. Irrigation water infiltration modeling using machine learning. *Comput. Electron. Agric.* **2020**, *180*, 105921. [CrossRef]
14. Akshay, S.; Ramesh, T.K. Efficient machine learning algorithm for smart irrigation. In Proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 28–30 July 2020; pp. 867–870.
15. Saranya, T.; Deisy, C.; Sridevi, S.; Anbananthen, K.S.M. A comparative study of deep learning and Internet of Things for precision agriculture. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106034. [CrossRef]
16. Abdallah, E.B.; Grati, R.; Boukadi, K. A machine learning-based approach for smart agriculture via stacking-based ensemble learning and feature selection methods. In Proceedings of the 2022 18th International Conference on Intelligent Environments (IE), Biarritz, France, 20–23 June 2022; pp. 1–8.
17. Lap, B.Q.; Phan, T.-T.; Du Nguyen, H.; Quang, L.X.; Hang, P.T.; Phi, N.Q.; Hoang, V.T.; Linh, P.G.; Hang, B.T.T. Predicting Water Quality Index (WQI) by feature selection and machine learning: A case study of An Kim Hai irrigation system. *Ecol. Inform.* **2023**, *74*, 101991. [CrossRef]

18. Barzegar, R.; Aalami, M.T.; Adamowski, J. Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model. *Stoch. Environ. Res. Risk Assess.* **2020**, *34*, 415–433. [CrossRef]
19. Zhao, L.; Gkountouna, O.; Pfoser, D. Spatial Auto-regressive Dependency Interpretable Learning Based on Spatial Topological Constraints. *ACM Trans. Spat. Algorithms Syst.* **2019**, *5*, 1–28. [CrossRef]
20. Bianchi, V.; Bassoli, M.; Lombardo, G.; Fornacciari, P.; Mordonini, M.; De Munari, I. IoT Wearable Sensor and Deep Learning: An Integrated Approach for Personalized Human Activity Recognition in a Smart Home Environment. *IEEE Internet Things J.* **2019**, *6*, 8553–8562. [CrossRef]
21. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image segmentation using deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3523–3542. [CrossRef] [PubMed]
22. Gupta, A.; Anpalagan, A.; Guan, L.; Khwaja, A.S. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array* **2021**, *10*, 100057. [CrossRef]
23. Zheng, Y.-Y.; Kong, J.-L.; Jin, X.-B.; Wang, X.-Y.; Su, T.-L.; Zuo, M. CropDeep: The Crop Vision Dataset for Deep-Learning-Based Classification and Detection in Precision Agriculture. *Sensors* **2019**, *19*, 1058. [CrossRef]
24. Moshayedi, A.J.; Roy, A.S.; Kolahdooz, A.; Shuxin, Y. Deep Learning Application Pros and Cons over Algorithm. *EAI Endorsed Trans. AI Robot.* **2022**, *1*, 1–13. [CrossRef]
25. Mathew, A.; Amudha, P.; Sivakumari, S. Deep learning techniques: An overview. In *Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020*; Springer: Singapore, 2021; pp. 599–608.
26. Kim, T.Y.; Cho, S.B. Predicting the household power consumption using CNN-LSTM hybrid networks. In *Proceedings of the Intelligent Data Engineering and Automated Learning–IDEAL 2018: 19th International Conference, Madrid, Spain, 21–23 November 2018*; Part I; Springer International Publishing: Cham, Switzerland, 2018; Volume 19, pp. 481–490.
27. Al-Shourbaji, I.; Kachare, P.H.; Abualigah, L.; Abdelhag, M.E.; Elnaim, B.; Anter, A.M.; Gandomi, A.H. A Deep Batch Normalized Convolution Approach for Improving COVID-19 Detection from Chest X-ray Images. *Pathogens* **2022**, *12*, 17. [CrossRef]
28. Scott, D.W. On optimal and data-based histograms. *Biometrika* **1979**, *66*, 605–610. [CrossRef]