Enhancing Neural Network Performance for Water Quality Forecasting with Principal Component Analysis in Intensive Aquaculture

Guilherme Freire de Almeida[1*], Lucas Cruz da Silva[2], Valéria Loureiro da Silva[3]
[1]Embedded Systems Department, [2]Robotics Department, [3]Optics and Photonics Laboratory; SENAI CIMATEC University; Salvador, Bahia, Brazil

This work investigates the application of Principal Component Analysis (PCA) to enhance the performance of a neural network regression model for water quality forecasting in intensive aquaculture. The dataset from an intensive cultivation study includes daily readings of controlled pond biochemical parameters. The standard and PCA-enhanced models had their performance evaluated based on the MSE, MAE, and R². The results demonstrate that the model incorporating PCA outperformed the standard model. The PCA model achieved lower training and testing MSEs, with a notable reduction in MAE. These findings highlight the effectiveness of PCA in improving the accuracy and efficiency of neural network models by reducing dimensionality and emphasizing the most informative features.

Keywords: Principal Component Analysis. Neural Network. Water Quality Forecasting.

In aquaculture, water quality forecasting has become an important area of research, especially for high-density cultivation methods such as semi-intensive and intensive systems. Predicting water quality parameters helps maintain optimal conditions for marine life, which is essential for sustainable aquaculture practices [1]. This approach often relies on artificial intelligence (AI) algorithms that use biochemical measurements to predict other critical parameters, such as dissolved oxygen and pH [2].

Principal Component Analysis (PCA) is a widely used statistical method for dimensionality reduction and feature extraction in neural network (NN) training datasets. This matrix-based technique transforms the data into a new feature space, retaining the essential characteristics while enabling a reduction in the number of features. This transformation can lead to more efficient models, particularly useful in scenarios with memory and processing constraints, as it enables a more compact and representative data representation [3].

This work explores the application of PCA in improving the performance of a regression neural network model for water quality forecasting. Specifically, it investigates two models: a typical neural network regression model that takes ammonia, nitrate, and temperature as inputs to predict dissolved oxygen and pH levels [4] and a second model that applies PCA to the input data before training the neural network. The objective is to assess whether PCA can enhance the model's efficiency and accuracy in predicting critical water quality parameters in intensive aquaculture systems.

Theoretical Background

Principal Component Analysis

Principal Component Analysis (PCA) is a technique used in dimensionality reduction. It transforms a large set of variables into a new, smaller set known as principal components. These principal components are linear combinations of the original variables and capture the most significant patterns in the data. The main objectives of PCA are to:

- **Preserve variance**: The method ensures that the new components retain as much of the original dataset's variability.
- **Eliminate redundancy**: PCA can reduce redundancy and highlight the most important

Received on 21 September 2024; revised 12 December 2024.
Address for correspondence: Guilherme Freire de Almeida. Avenida Orlando Gomes, 1845. Zipcode: 41650-010. Salvador, Bahia, Brazil. Zipcode: 40170-115. E-mail: freireg1503@gmail.com.

features by identifying and combining correlated variables.

In addition to dimensionality reduction, PCA can be used for feature extraction. Feature extraction through PCA transforms data into a new set of features that more accurately represent the underlying structure by emphasizing components that capture the most variance. This process reduces noise and highlights the most important aspects, enhancing machine learning models' performance.

$$\boldsymbol{P} = \boldsymbol{KX} \tag{1}$$

The method achieves this through a linear transformation (Eq. 1), where K is a transformation matrix obtained by diagonalizing the covariance matrix of the original data set X. The new representation, P, retains the same dimensionality as the original data but orders the components by their contribution to total variance [5]. Typically, only the first few principal components are retained as they explain most of the variance, allowing the less significant components to be discarded. This reduces the dimensionality compared with the original representation and decreases the complexity of subsequent classification systems [6].

Neural Networks

Artificial Neural Networks, or simply Neural Networks (NN), are machine learning models designed to recognize patterns and make predictions inspired by the human brain's structure and function. These networks consist of interconnected layers of nodes, known as neurons, that process input data and generate outputs based on learned relationships within the data [3].

A basic neural network comprises three main types of layers:

- **Input Layer**: The input layer receives the raw data that the network will process. Each node in this layer represents a different feature or variable from the dataset. In this layer, no data computation is done.
- **Hidden Layer**: The hidden layers are situated between the input and output layers and perform most of the network's computation. These layers consist of neurons that apply weighted transformations to the input data, followed by an activation function. NNs can have multiple hidden layers.
- **Output Layer**: As the name suggests, the output layers represent the last layer in the network, producing the prediction output.

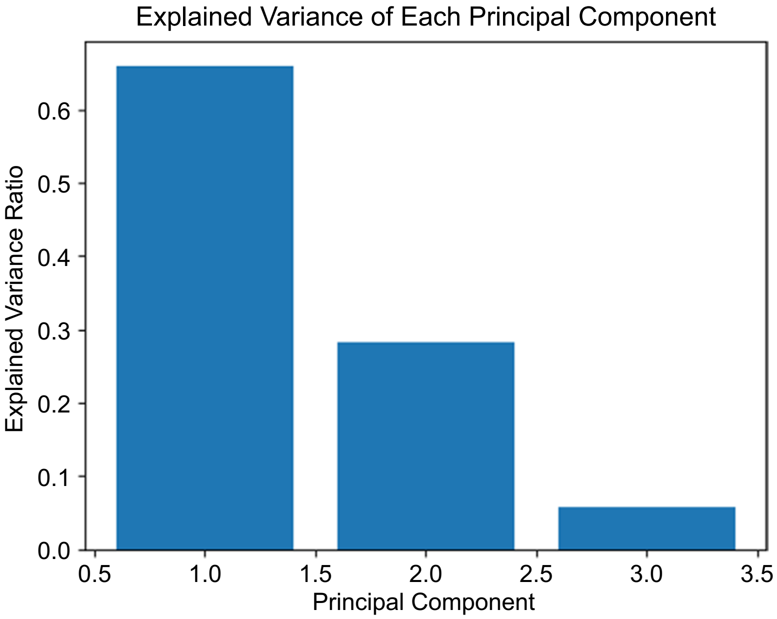
During training, the neural network learns by adjusting the weights associated with each connection between neurons. This is achieved through a process known as backpropagation, where the model's prediction error is propagated backward through the network, updating the weights to minimize the error. This iterative process continues until the network's predictions are sufficiently accurate [7].

Materials and Methods

The data used for this demonstration were collected from an intensive tilapia cultivation study at the aquaculture research site of Fazenda Oruabo in Santo Amaro, Bahia. The cultivation occurred from July 30th, 2018, to December 11th of the same year in a controlled tank exposed to sunlight, with artificial oxygenation and a regular feeding schedule. The dataset includes daily readings of dissolved oxygen, pH, temperature, ammonia, and nitrite at 8:00 am each morning. The dataset had 134 samples with a 70%, 15%, and 15% split for training, validation, and testing, respectively.

Initially, a network was designed to take raw temperature, ammonia, and nitrite samples as input to forecast oxygen and pH values. Following this, a new model will achieve the same forecasting task and be trained using the same raw data but preprocessed with PCA. Figure 1 shows the component analysis referring to the object in evaluation, where the first component represents 65.98% of the total variance, the second one is 28.25%, and the last is only 5.77%. Removing the third component would not severely

Figure 1. Explained variance for each component.



disturb the system's variance in that context. This is important during the NN training, where, after the preprocessing, this component could be removed to reduce the model's input dimensions.

The trained models were feedforward neural networks with four hidden layers, each consisting of 128 and two output nodes. The ReLU (Rectified Linear Unit) activation function was used, and the Adam optimizer was employed for backpropagation with a 0.001 learning rate.

The models' training performance was evaluated using mean squared error (MSE) as the loss function and mean absolute error (MAE) as a secondary metric. In contrast, their real-world performance was assessed using MSE and the coefficient of determination (R^2).

MSE measures the average squared difference between the predicted and actual values, indicating how close the model's predictions are to the valid values. R^2 represents the proportion of the variance in the dependent variable explained by the model's independent variables. A higher R^2 value indicates that the model explains a more significant portion of the variability in the data, demonstrating a better fit [8].

All the development was executed on Google's Colab platform, utilizing a cloud-based Linux environment with 12GB of RAM, and implemented using TensorFlow and the Keras API.

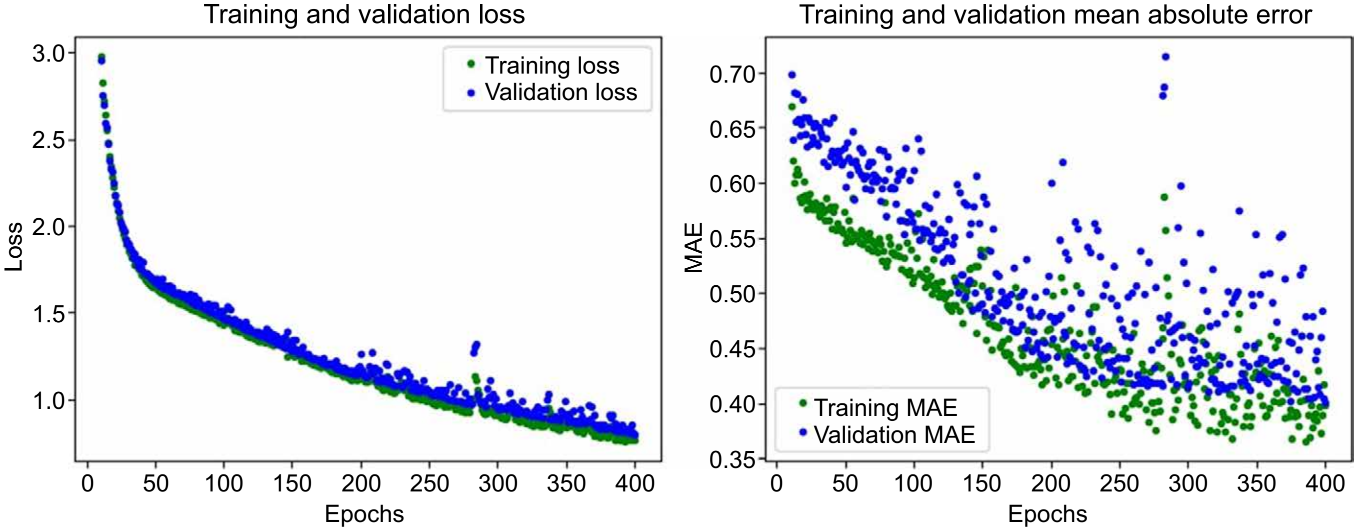
Results and Discussion

The standard neural network regression model and the model with PCA-applied inputs were trained for 400 epochs as performance improvements plateaued beyond this point. The training history illustrates the progression of training loss, validation loss, training mean absolute error, and validation MAE throughout the training process.

Analysis of Training and Validation Loss

Figure 2 shows that the training and validation loss decreased significantly during the initial epochs and stabilized as training progressed. The training loss converged to around 0.78, while the validation loss reached a similar level. The gap between training and validation loss suggests that the model generalizes well to unseen data with minimal overfitting. The training and validation MAE followed a similar trend to the loss values, with both metrics decreasing over time. The training MAE decreased steadily to approximately 0.40, while the validation

Figure 2. Model training Loss and MAE.



MAE also showed a steady decline, indicating good predictive performance.

In Figure 3, the training and validation loss started at higher values than the standard model, reflecting the initial higher error. However, both losses decreased rapidly and converged to lower values, around 0.38, suggesting better performance. The close superposition of the training and validation loss curves indicates minimal overfitting and good generalization. The training and validation MAE also started at higher values than the first model but presented a faster stabilization profile. Both metrics reached approximately 0.26, with the validation MAE closely following the training MAE curve, further indicating strong model performance and generalization.

Analysis of Testing Results

Table 1 summarizes the test performance metrics for both the standard neural network model (without PCA) and the model incorporating PCA.

Model Training Metrics

- **Without PCA**: The model achieved a training MSE of 0.7849 and a training MAE of 0.4042.
- **With PCA**: The PCA model demonstrated improved training performance, with a significantly lower training MSE of 0.3887 and MAE of 0.2605.

Testing Metrics for Oxygen Prediction

- **Without PCA**: The testing MSE for dissolved oxygen was 0.3998, with an R^2 value of 0.0248, indicating poor predictive accuracy and low variance explanation.
- **With PCA**: The PCA model achieved a notably lower testing MSE of 0.2148 for oxygen prediction, with an R^2 value of 0.4758. This substantial increase in R^2 indicates that the PCA model discriminates more of the variance in the oxygen data, leading to more accurate predictions.

Testing Metrics for pH Prediction

- **Without PCA**: The testing MSE for pH prediction was 0.1551, with an R^2 value of 0.0601.
- **With PCA**: The PCA model showed a significant improvement, with a testing MSE of 0.0562 and an R^2 value of 0.6591. This indicates a much better fit and higher accuracy in predicting pH levels.

Figure 4 and Figure 5 display the test outputs for both scenarios, showing that the enhanced

Figure 3. Model training Loss and MAE with PCA.

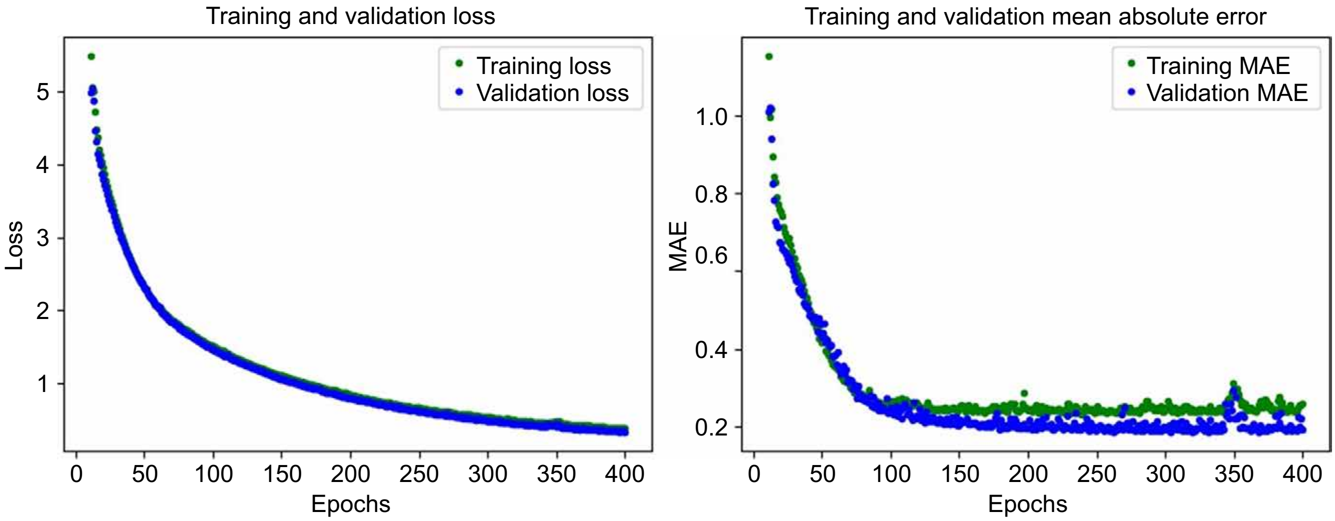


Table 1. Results comparison.

	Standard Model	PCA Model
Model Training MSE	0.7849	0.3887
Model Training MAE	0.4042	0.2605
Oxygen Testing MSE	0.3998	0.2148
Oxygen Testing R²	0.0248	0.4758
pH Testing MSE	0.1551	0.0562
pH Testing R²	0.0601	0.6591

Figure 4. Standard model test results.

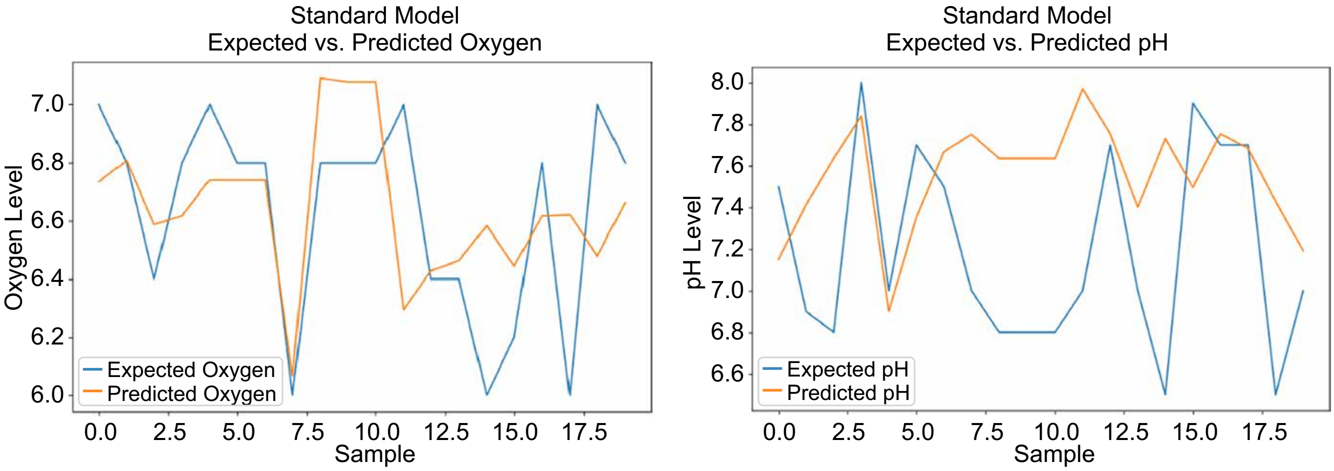
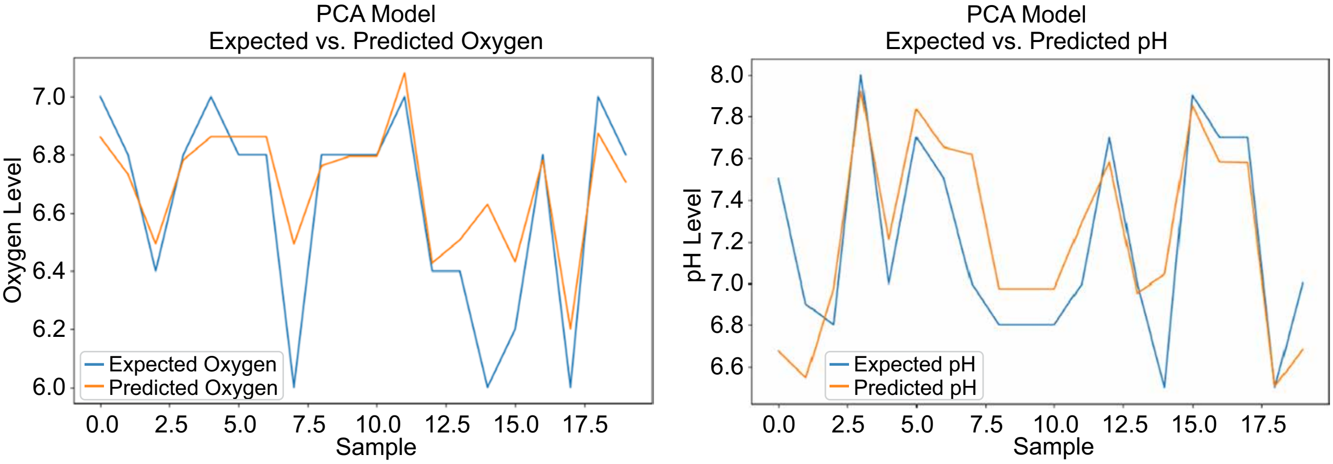


Figure 5. PCA model test results.



model provides a better fit. This is evidenced by the prediction graph shapes, which resemble the expected output more closely than the standard model.

Comparative Analysis and Discussion

The PCA-applied model significantly reduced training and validation losses more than the standard model. The minimal gap between training and validation losses suggests that the PCA model generalizes better, likely due to the reduced dimensionality and elimination of redundant information, which helps mitigate overfitting.

The test MSE and R^2 metrics confirmed the superior performance of the PCA model when compared to the standard model, with lower errors and a higher proportion of explained variance. This indicates that PCA effectively enhanced the model's ability to predict dissolved oxygen and pH levels from the input parameters.

Conclusion

The findings from this work underscore the importance of feature extraction techniques like PCA in enhancing the performance of neural network models, even in non-high-dimensional datasets like the one presented. By reducing noise and emphasizing the most informative aspects of the data, the method has proven to be a valuable tool in improving both the accuracy and efficiency of the built predictive model compared with its standard counterpart.

Moreover, investigating the impact of these techniques in different aquaculture environments and with various water quality parameters could provide more generalizable insights. It would also be beneficial to explore real-time implementation and the computational trade-offs associated with these advanced preprocessing methods, particularly in resource-constrained settings.

References

1. Yu C et al. Water Quality Management in Intensive Aquaculture in China. Springer, p. 1243–1252, 2008.
2. Zhao S et al. Application of machine learning in intelligent fish aquaculture: A review. Aquaculture 2021;540:736724.
3. Haykin SS. Neural Networks. Upper Saddle River, N.J.: Prentice Hall, 1999.
4. Han H-G, Chen Q, Qiao J-F. An efficient self-organizing RBF neural network for water quality prediction. Neural Networks 2011;24(7):717–725.
5. Silva LC et al. Embedded decision support system for ultrasound nondestructive evaluation based on extreme learning machines. Computers & Electrical Engineering 2021;90:106891.
6. Ferrani DG, Nunes L. Introdução a mineração de dados. [s.l.] Saraiva Educação S.A., 2017.
7. Aggarwal CC. Neural Networks and Deep Learning.
8. Li T et al. Predicting aquaculture water quality using machine learning approaches. Water 2022;14(18):2836.