

Exploring the Role of Hydrological Parameters in *Labeo rohita* Growth Using Python Program

Nagaraj Narayan Sannabhadti¹, Sana Ahmed², Sufia Zaman³, Subhra Bikash Bhattacharyya⁴, and Abhijit Mitra⁵

¹Research Scholar, Department of Oceanography, Techno India University, West Bengal, EM 4 Salt Lake, Sector V, Kolkata 700091, India.

²Assistant Professor, Department of Oceanography, Techno India University, West Bengal, EM 4 Salt Lake, Sector V, Kolkata 700091, India.

³Associate Professor, Department of Oceanography, Techno India University, West Bengal, EM 4 Salt Lake, Sector V, Kolkata 700091, India.

⁴Adjunct Professor, Department of Oceanography, Techno India University, West Bengal, EM 4 Salt Lake, Sector V, Kolkata 700091, India.

⁵Director (Research), Techno India University, West Bengal, EM 4 Salt Lake, Sector V, Kolkata 700091, India.

Abstract

This study highlights the efficacy of Python program in carp aquaculture by delivering precise and efficient growth assessments through inputs of length and weight of the cultured species. The condition index of the cultured species, *Labeo rohita*, showed variations across the three ponds, P₁, P₂, and P₃. By addressing the hydrological parameters impacting pond performance and harnessing Artificial Intelligence, aquaculture practices can be optimized for greater productivity and sustainability. Future research should aim to improve the accuracy of AI models and explore new applications to further advance aquaculture practice.

Keywords: *Labeo rohita*, Condition Index, hydrological parameters, Python program

1. Introduction

The aquaculture industry, particularly in developing countries, has experienced significant growth over the past few decades. This growth is largely driven by the increasing demand for fish as a primary source of protein for a growing global population. Among various aquaculture species, *Labeo rohita*, commonly known as rohu, is one of the most widely cultivated freshwater fish species in South Asia. Its popularity stems from its high nutritional value, economic importance, and adaptability to diverse environmental conditions (Jhingran, 1991). However, optimizing the growth and health of *Labeo rohita* in aquaculture systems remains a challenging task due to the complex interplay of various environmental factors, particularly hydrological parameters of the culture ponds.

Hydrological parameters, including water temperature, dissolved oxygen levels, pH, and nutrient level (preferably nitrate and phosphate), play a crucial role in influencing the growth,

health, and overall productivity of carp in aquaculture systems. These parameters directly affect the physiological processes of fish, such as metabolism, respiration, and immune responses (Boyd, 1998). Therefore, maintaining optimal hydrological conditions is essential for achieving sustainable and efficient aquaculture practices. Traditional methods of monitoring and managing these parameters often rely on manual measurements and empirical knowledge, which can be time-consuming, labour-intensive, and prone to human error.

In recent years, advancements in technology and data science have opened new avenues for enhancing aquaculture practices (Patil et al., 2019; Sharma et al., 2021). Machine learning (ML), a subset of artificial intelligence (AI), has emerged as a powerful tool for analyzing complex datasets and extracting meaningful insights. Machine learning algorithms can process vast amounts of data, identify patterns, and make accurate predictions, thereby offering significant potential for optimizing aquaculture operations (Chen et al., 2020). By leveraging machine learning techniques, it is possible to develop predictive models that can forecast the growth and health of *Labeo rohita* (using Condition Index abbreviated as CI as proxy) based on relevant hydrological parameters of the culture pond. These models can help aquaculturists take decisions regarding water quality management, feeding regimes, and other critical aspects of fish farming.

The application of machine learning in aquaculture is not entirely new. Several studies have demonstrated the feasibility and effectiveness of using machine learning algorithms for various purposes, such as disease detection, water quality prediction, and feed optimization (Jiang et al., 2021). However, the specific application of machine learning to optimize the growth of *Labeo rohita* by analyzing the hydrological parameters remains an underexplored area. This study aims to fill this gap by investigating the potential of Python program to monitor the CI values of the species during the culture tenure of 210 days during 2023.

The first step in this endeavour is to collect and preprocess relevant data on hydrological parameters and fish growth metrics. Data collection involves continuous monitoring of water quality parameters using sensors and regular measurements of fish growth indicators, such as weight and length (Mishra et al., 2018). Preprocessing the data includes cleaning, normalization, and transformation to ensure it is suitable for analysis by machine learning algorithms. Once the data is ready, various machine learning algorithms, can be applied to identify the relationships between hydrological parameters and fish growth.

The integration of machine learning models with real-time monitoring systems can further enhance the practical utility of this approach. By deploying sensors and IoT (Internet of Things) devices in aquaculture ponds, it is possible to collect real-time data on hydrological parameters and feed it into machine learning models. These models can then provide real-time predictions and recommendations to aquaculturists, enabling proactive management of water quality and other environmental factors (Wang et al., 2019). For instance, if the model predicts a decline in dissolved oxygen levels that could negatively affect fish growth, appropriate aeration measures can be implemented promptly.

Moreover, the use of machine learning in aquaculture is not limited to optimizing growth. It can also contribute to sustainability and environmental conservation. By analyzing data on water quality and fish health, machine learning models can help identify the water quality of the cultured pond, assess the impact of hydrological parameters on the cultured species, and develop strategies for mitigating adverse effects (Zhang et al., 2020). This holistic approach aligns with the principles of sustainable aquaculture, which aim to balance economic productivity with environmental stewardship.

In addition to technical and environmental benefits, the application of machine learning in aquaculture also has significant economic implications. By optimizing growth and improving the efficiency of aquaculture operations, machine learning can enhance profitability for fish farmers. Reduced mortality rates, better feed conversion ratios, and improved fish health translate into higher yields and lower operational costs (García et al., 2017). This is particularly important for small-scale farmers who rely on carp culture as their primary source of livelihood.

Despite the promising potential of machine learning in aquaculture, several challenges need to be addressed to ensure its successful implementation. Data quality and availability are critical factors, as accurate and comprehensive data is essential for training reliable machine learning models. The integration of sensors and IoT devices in aquaculture ponds requires significant investment and technical expertise. Additionally, the adoption of machine learning technologies in aquaculture necessitates training and capacity-building for farmers and aquaculturists to ensure they can effectively use and interpret the outputs of machine learning models (Khan & Khan, 2019).

Thus, the application of machine learning algorithms for optimizing the growth of *Labeo rohita* in controlled aquaculture environments holds great promise. By leveraging the power of machine learning to analyse hydrological parameters, it is possible to develop predictive models that enhance growth assessment, improve water quality management, and contribute to sustainable aquaculture practices. This study aims to explore this potential and provide valuable insights into the role of hydrological parameters in fish growth, paving the way for more efficient and sustainable aquaculture systems.

2. Methodology

2.1 Experimental Setup and Monitoring of Hydrological Parameters

In this study, three ponds were designated for the experiment as shown in Fig.1. Traditional feed was provided twice daily, once in the morning and once in the evening, ensuring that the feed quantity was adjusted based on the biomass of the fish to prevent overfeeding and maintain water quality. This feeding regime continued from March to October during 2023. Regular monitoring of water quality parameters such as surface water temperature, surface water pH, dissolved oxygen, dissolved nitrate, and dissolved phosphate levels was carried out to ensure optimal conditions for the carp's growth and health. Soil Organic Carbon (SOC) of the pond bottom was also analyzed to evaluate the pond bottom environment that often plays important role in maintaining the pond environment.

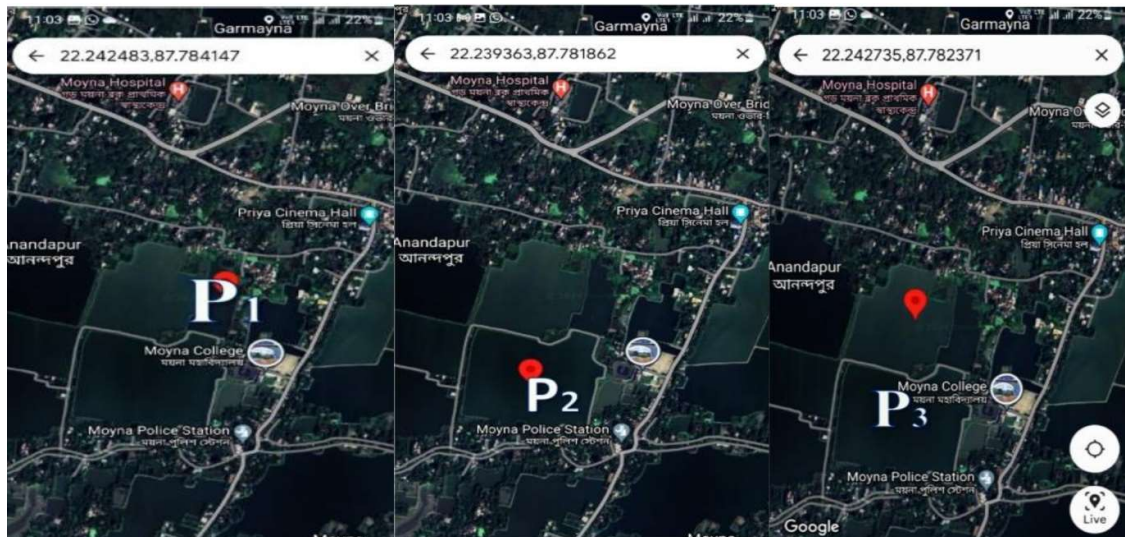


Fig. 1. Location of ponds at Moyna in the East Midnapur district of West Bengal

Fish CI was assessed monthly by measuring the weight and length of a random sample of 23-25 fishes from each pond. Health status of the sampled fishes was evaluated through visual inspection for any signs of disease or abnormalities. At the end of the seven-month experimental period (*i.e.*, October), the final weight and length of the fish were measured, and a comparative analysis was performed to evaluate the CI of the cultured species.

2.2 Monitoring of Hydrological Parameters

Water samples for hydrological parameters were collected from the surface using a clean bucket. Dissolved oxygen samples were directly collected in 150 ml BOD bottles without any agitation and immediately fixed post-collection. Temperature and pH were measured on-site, while additional water samples were collected from each pond in Tarson bottles and transported to the laboratory at 4°C for nutrient analysis. Hydrological parameters were analyzed from the three selected ponds at Moyna.

Surface water temperature: Measured using a mercury Celsius thermometer ranging from 0° to 100°C.

Surface water pH: Determined using a portable pH meter with a sensitivity of ±0.02, calibrated with pH buffers 4.0 and 7.0 before each use.

Dissolved oxygen: Initially measured in the field with a DO meter and then confirmed in the laboratory using Winkler’s method (Strickland & Parsons, 1972).

Dissolved inorganic nutrients: Surface water for dissolved nutrient analysis was collected in clean Tarson bottles and transported to the lab in ice-cooled conditions. Triplicate samples ensured data quality. Nutrient concentrations in surface water were determined using the standard spectrophotometric method by Strickland and Parsons (1972). Nitrate was reduced to nitrite by passing the sample through a glass column with ammonium chloride buffer and

amalgamated cadmium filings. The nitrite was then treated with sulphanilamide to form a diazonium ion, which was coupled with N-1-naphthyl ethylenediamine to produce a pink azo dye. Phosphate was measured by treating a sample aliquot with an acidic molybdate reagent containing ascorbic acid and a small amount of potassium antimony tartrate.

Soil Organic Carbon (SOC): Soil samples from the top 5 cm were collected from each pond and dried at 60°C for 48 hours. During preparation, visible plant particles were handpicked and removed from the soil. The soil was then sieved through a 2 mm mesh. A 50-gram sample of bulk soil from each pond was finely ground using a ball mill. The finely ground samples were then randomly mixed to create a representative sample of the selected pond. The soil organic carbon (SOC) percentage was determined using a modified Walkley and Black method (1934). This meticulous approach to monitoring and maintaining water quality aligns with the broader goals of sustainable aquaculture practices and can significantly contribute to optimizing the health and growth of *Labeo rohita*, much like how advancements in AI and machine learning are transforming various sectors by enhancing efficiency and sustainability.

2.3 Python Program to Evaluate Pond-wise Condition Index of *Labeo rohita*

The code provided here is a Python program that uses the tkinter library to create a graphical user interface (GUI). This program is designed to manage and analyse data related to the condition of fish in various ponds. It includes functionalities for inputting data, calculating a condition index for the fish, and visualizing this data through different types of plots (heatmaps and contour maps). We have not displayed here the visualizing of the data through different types of plots (heatmaps and contour maps).

Here is a breakdown of its main components and functionalities:

1. **Data Handling:** It uses pandas to manage data related to fish conditions in various ponds. Data such as date, pond name, area, stocking density, survival rate, average body weight, and body length are stored and manipulated.
2. **Calculations:** The program includes a function to calculate the condition index of fish, which is an indicator of fish health based on their weight and length.
3. **Visualization:** It uses matplotlib and seaborn for visualizing the data through heatmaps and contour maps, which help to analyze the spatial distribution of fish condition indices over time and across different ponds.
4. **GUI Components:** tkinter is used for the GUI, allowing users to input data, view results in a structured format, and interact with data visualizations through various controls like buttons and entry fields.
5. **Geographical Conversion:** The program includes a function to convert degrees, minutes, and seconds to decimal degrees, which is useful for plotting geographical data.

CONDITION INDEX OF FISH USING PYTHON

```
import tkinter as tk
from tkinter import messagebox, Toplevel, ttk
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
# Initialize an empty DataFrame to store results
results_df = pd.DataFrame(columns=[
    'Date', 'Pond Name', 'Area (m^2)', 'Stocking Density (/m^2)', 'Total Stocking',
    'Survival Rate (%)', 'Total Live Population', 'Average Body Weight (g)',
    'Average Body Length (cm)', 'Condition Index', 'Latitude', 'Longitude'
])
def dms_to_dd(degrees, minutes, seconds):
    return degrees + minutes / 60 + seconds / 3600
def calculate_condition_index(date, pond_name, area, stocking_density, survival_rate, avg_body_weight,
    avg_body_length, latitude, longitude):
    global results_df
    # Calculate total stocking
    total_stocking = stocking_density * area
    # Calculate total live population of fish
    total_live_population = total_stocking * (survival_rate / 100)
    # Calculate condition index
    condition_index = (avg_body_weight / (avg_body_length ** 3)) * 100
    # Add the calculation values to the DataFrame
    new_row = pd.DataFrame({
        'Date': [date], 'Pond Name': [pond_name], 'Area (m^2)': [area],
        'Stocking Density (/m^2)': [stocking_density], 'Total Stocking': [total_stocking],
        'Survival Rate (%)': [survival_rate], 'Total Live Population': [total_live_population],
        'Average Body Weight (g)': [avg_body_weight], 'Average Body Length (cm)': [avg_body_length],
        'Condition Index': [condition_index], 'Latitude': [latitude], 'Longitude': [longitude]
    })
    results_df = pd.concat([results_df, new_row], ignore_index=True)
    return total_stocking, total_live_population, condition_index
def show_popup_chart(fig):
    # Create a new Toplevel window
```

```
popup = Toplevel()
popup.wm_title("Chart")
# Create a canvas to display the matplotlib figure
canvas = FigureCanvasTkAgg(fig, master=popup)
canvas.draw()
canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=1)
# Add a button to close the popup
close_button = tk.Button(popup, text="Close", command=popup.destroy)
close_button.pack()
def plot_heatmap():
    if results_df.empty:
        messagebox.showinfo("No Data", "No data available to plot.")
        return
    # Pivot the DataFrame for heatmap
    heatmap_data = results_df.pivot(index='Pond Name', columns='Date', values='Condition Index')
    fig, ax = plt.subplots(figsize=(12, 8))
    sns.heatmap(heatmap_data, annot=True, cmap="YlGnBu", linewidths=.5, cbar_kws={'label': 'Condition
Index'}, ax=ax)
    ax.set_title('FISH CONDITION INDEX HEAT MAP')
    ax.set_xlabel('Date')
    ax.set_ylabel('Pond Name')
    show_popup_chart(fig)
def show_contour_map():
    if results_df.empty:
        messagebox.showinfo("No Data", "No data available to plot.")
        return
    # Create a new Toplevel window for date selection
    date_selection_popup = Toplevel()
    date_selection_popup.wm_title("Select Date(s)")
    date_label = tk.Label(date_selection_popup, text="Select Date(s):")
    date_label.pack(padx=10, pady=5)
    # Listbox to select multiple dates
    date_listbox = tk.Listbox(date_selection_popup, selectmode=tk.MULTIPLE)
    for date in results_df["Date"].unique():
        date_listbox.insert(tk.END, date)
    date_listbox.pack(padx=10, pady=5)
```

```

def generate_contour_map():
    selected_dates = [date_listbox.get(idx) for idx in date_listbox.curselection()]
    if not selected_dates:
        messagebox.showinfo("No Selection", "No date selected.")
        return
    for selected_date in selected_dates:
        fig, ax = plt.subplots(figsize=(12, 8))
        # Filter data by selected date
        date_filtered_df = results_df[results_df['Date'] == selected_date]
        # Extract latitude, longitude, and condition index from the filtered DataFrame
        latitudes = date_filtered_df['Latitude']
        longitudes = date_filtered_df['Longitude']
        condition_indices = date_filtered_df['Condition Index']
        # Plot KDE (Kernel Density Estimation) plot
        sns.kdeplot(
            x=longitudes, y=latitudes,
            weights=condition_indices, cmap="viridis", shade=True, ax=ax
        )
        ax.set_title(f'VARIATION MAP OF CONDITION INDEX OF FISH ({selected_date})')
        ax.set_xlabel('Longitude')
        ax.set_ylabel('Latitude')
        # Add pond names as annotations
        for _, row in date_filtered_df.iterrows():
            ax.annotate(row['Pond Name'], (row['Longitude'], row['Latitude']), textcoords="offset points",
                xytext=(5,-5), ha='center')
        show_popup_chart(fig)
        date_selection_popup.destroy()
        generate_button = tk.Button(date_selection_popup, text="Generate Contour Map",
            command=generate_contour_map)
        generate_button.pack(pady=10)
def show_data():
    if results_df.empty:
        messagebox.showinfo("No Data", "No data available.")
        return
def edit_row():
    selected_item = tree.selection()
    if not selected_item:

```



```
messagebox.showinfo("No Selection", "No row selected.")

return

item = tree.item(selected_item)
values = item['values']

edit_popup = Toplevel()
edit_popup.wm_title("Edit Data")

labels = [
    "Date (YYYY-MM-DD):", "Pond Name:", "Area (m^2):", "Stocking Density (/m^2):",
    "Survival Rate (%):", "Average Body Weight (g):", "Average Body Length (cm):",
    "Latitude (Deg Min Sec):", "Longitude (Deg Min Sec):"
]

entries = []
for i, text in enumerate(labels):
    label = tk.Label(edit_popup, text=text)
    label.grid(row=i, column=0, padx=10, pady=5)

    entry = tk.Entry(edit_popup)
    entry.grid(row=i, column=1, padx=10, pady=5)

    entry.insert(0, values[i])
    entries.append(entry)

def save_edits():
    for i, entry in enumerate(entries):
        tree.set(selected_item, column=i, value=entry.get())

    edit_popup.destroy()

save_button = tk.Button(edit_popup, text="Save", command=save_edits)
save_button.grid(row=len(labels), column=0, colspan=2, pady=10)

# Create a new Toplevel window to show the DataFrame
data_popup = Toplevel()
data_popup.wm_title("Data")

tree = ttk.Treeview(data_popup, columns=list(results_df.columns), show='headings')

for col in results_df.columns:
    tree.heading(col, text=col)
    tree.column(col, width=100)

for row in results_df.itertuples(index=False):
    tree.insert("", tk.END, values=row)

tree.pack(fill=tk.BOTH, expand=True)

# Add buttons to close and edit the popup
```

```

close_button = tk.Button(data_popup, text="Close", command=data_popup.destroy)
close_button.pack(side=tk.LEFT, padx=10, pady=10)

edit_button = tk.Button(data_popup, text="Edit Data", command=edit_row)
edit_button.pack(side=tk.RIGHT, padx=10, pady=10)

def submit():
    date = entry_date.get()
    pond_name = entry_pond_name.get()
    area = float(entry_area.get())
    stocking_density = float(entry_stocking_density.get())
    survival_rate = float(entry_survival_rate.get())
    avg_body_weight = float(entry_avg_body_weight.get())
    avg_body_length = float(entry_avg_body_length.get())

    # Parse latitude and longitude in degrees, minutes, seconds format
    lat_deg, lat_min, lat_sec = map(float, entry_latitude.get().split())
    lon_deg, lon_min, lon_sec = map(float, entry_longitude.get().split())

    latitude = dms_to_dd(lat_deg, lat_min, lat_sec)
    longitude = dms_to_dd(lon_deg, lon_min, lon_sec)

    total_stocking, total_live_population, condition_index = calculate_condition_index(
        date, pond_name, area, stocking_density, survival_rate, avg_body_weight, avg_body_length, latitude,
        longitude
    )

    if condition_index < 1.2:
        suggestion = "The condition index is too low. Consider improving nutrition and reducing stress factors."
    elif 1.2 <= condition_index <= 2.0:
        suggestion = "The condition index is within the optimal range. Maintain current management practices."
    else:
        suggestion = "The condition index is too high. Monitor for potential issues such as overfeeding or
        environmental factors."

    # Create a popup to show the result
    result_popup = Toplevel()
    result_popup.wm_title("Condition Index Result")

    result_label = tk.Label(result_popup, text=f"Condition Index: {condition_index:.2f}\nTotal Stocking:
    {total_stocking}\nTotal Live Population: {total_live_population}\nSuggestion: {suggestion}")
    result_label.pack(pady=10)

    close_button = tk.Button(result_popup, text="Close", command=result_popup.destroy)
    close_button.pack(pady=10)

# Create the main window

```

```
root = tk.Tk()
root.title("FISH CONDITION INDEX")
# Create and place the input fields
labels = [
    "Latitude (Deg Min Sec):", "Longitude (Deg Min Sec):",
    "Date (YYYY-MM-DD):", "Pond Name:", "Area (m^2):", "Stocking Density (/m^2):",
    "Survival Rate (%)":, "Average Body Weight (g):", "Average Body Length (cm):"
]
for i, text in enumerate(labels):
    label = tk.Label(root, text=text)
    label.grid(row=i, column=0, padx=10, pady=5)
entry_latitude = tk.Entry(root)
entry_latitude.grid(row=0, column=1, padx=10, pady=5)
entry_longitude = tk.Entry(root)
entry_longitude.grid(row=1, column=1, padx=10, pady=5)
entry_date = tk.Entry(root)
entry_date.grid(row=2, column=1, padx=10, pady=5)
entry_pond_name = tk.Entry(root)
entry_pond_name.grid(row=3, column=1, padx=10, pady=5)
entry_area = tk.Entry(root)
entry_area.grid(row=4, column=1, padx=10, pady=5)
entry_stocking_density = tk.Entry(root)
entry_stocking_density.grid(row=5, column=1, padx=10, pady=5)
entry_survival_rate = tk.Entry(root)
entry_survival_rate.grid(row=6, column=1, padx=10, pady=5)
entry_avg_body_weight = tk.Entry(root)
entry_avg_body_weight.grid(row=7, column=1, padx=10, pady=5)
entry_avg_body_length = tk.Entry(root)
entry_avg_body_length.grid(row=8, column=1, padx=10, pady=5)
# Create and place the buttons
submit_button = tk.Button(root, text="Submit", command=submit)
submit_button.grid(row=9, column=0, padx=10, pady=10)
data_button = tk.Button(root, text="Show Data", command=show_data)
data_button.grid(row=9, column=1, padx=10, pady=10)
contour_button = tk.Button(root, text="VARIATION MAP OF CONDITION INDEX OF FISH",
command=show_contour_map)
```

```
contour_button.grid(row=10, column=1, padx=10, pady=10)
# Run the main loop
root.mainloop()
```

3. Results

3.1 Condition Index

The results indicated that the machine learning models provided accurately the value of condition index for three selected ponds (Figs. 2-4). The condition factor (K) values indicate better health and well-being of the fish in the culture pond. Higher K values indicate better health and growth conditions for the carp. The results of our computations are presented here.

- **Pond P₁**: The K value indicated robust growth conditions, reflecting good health and well-being of the carp. The condition index is highest in this pond with a value of 1.97.
- **Pond P₂**: The K value of this pond is 1.35, which is lowest amongst the three ponds suggesting stressful environment.
- **Pond P₃**: The K value of this pond is 1.60, which indicate that the environment is better than pond P₂.

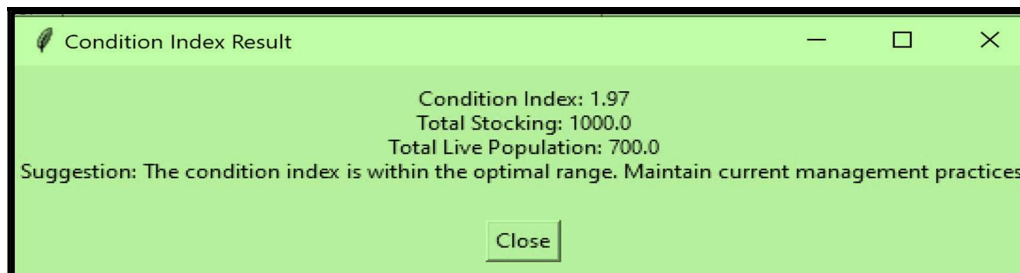


Fig. 2 Condition Index of *Labeo rohita* in Pond P₁ based on ABW and ABL during October 2023

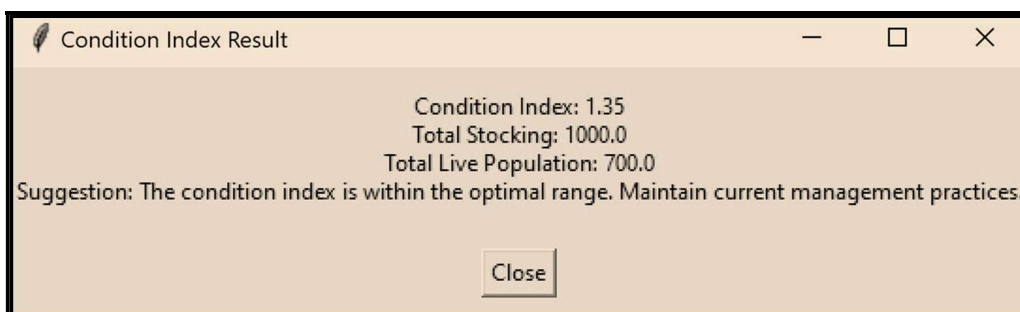


Fig. 3 Condition Index of *Labeo rohita* in Pond P₂ based on ABW and ABL during October 2023

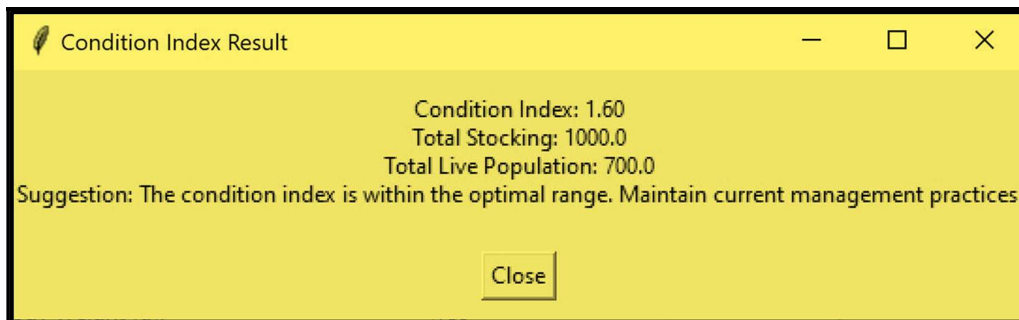


Fig. 4 Condition Index of *Labeo rohita* in Pond P₃ based on ABW and ABL during October 2023

3.2. Hydrological Parameters

The relevant hydrological parameters of three ponds along with the soil organic carbon is presented in the Tables 1-8.

Table 1. Water and soil quality of four selected ponds during March 2023

Parameter	P ₁	P ₂	P ₃
Surface water temperature (°C)	34.3	34.3	34.3
Surface water pH	7.9	7.5	7.7
Dissolved oxygen (ppm)	5.94	5.46	5.73
Dissolved Nitrate (µgm at l ⁻¹)	22.02	19.93	21.33
Dissolved Phosphate (µgm at l ⁻¹)	1.67	0.98	1.22
Soil Organic Carbon (%)	1.41	2.08	1.91

Table 2. Water and soil quality of four selected ponds during April 2023

Parameter	P ₁	P ₂	P ₃
Surface water temperature (°C)	34.6	34.5	34.4
Surface water pH	7.9	7.4	7.8
Dissolved oxygen (ppm)	5.93	5.49	5.71
Dissolved Nitrate (µgm at l ⁻¹)	22.24	20.24	21.63
Dissolved Phosphate (µgm at l ⁻¹)	1.68	1.04	1.27
Soil Organic Carbon (%)	1.33	2.01	1.82

Table 3. Water and soil quality of four selected ponds during May 2023

Parameter	P ₁	P ₂	P ₃
Surface water temperature (°C)	36.5	36.3	36.5
Surface water pH	7.9	7.5	7.7
Dissolved oxygen (ppm)	5.97	5.52	5.77
Dissolved Nitrate (µgm at l ⁻¹)	22.62	20.51	21.72
Dissolved Phosphate (µgm at l ⁻¹)	1.74	1.06	1.29
Soil Organic Carbon (%)	1.31	1.97	1.79

Table 4. Water and soil quality of four selected ponds during June 2023

Parameter	P ₁	P ₂	P ₃
Surface water temperature (°C)	39.6	39.6	39.4
Surface water pH	7.8	7.6	7.7
Dissolved oxygen (ppm)	5.99	5.68	5.78
Dissolved Nitrate (µgm at l ⁻¹)	22.76	20.70	22.12
Dissolved Phosphate (µgm at l ⁻¹)	1.76	1.10	1.33
Soil Organic Carbon (%)	1.28	1.93	1.72

Table 5. Water and soil quality of four selected ponds during July 2023

Parameter	P ₁	P ₂	P ₃
Surface water temperature (°C)	39.7	39.7	39.6
Surface water pH	7.9	7.5	7.6
Dissolved oxygen (ppm)	6.03	5.69	5.79
Dissolved Nitrate (µgm at l ⁻¹)	23.20	21.07	22.31
Dissolved Phosphate (µgm at l ⁻¹)	1.78	1.14	1.37
Soil Organic Carbon (%)	1.23	1.89	1.69

Table 6. Water and soil quality of four selected ponds during August 2023

Parameter	P ₁	P ₂	P ₃
Surface water temperature (°C)	37.2	37.1	37.0
Surface water pH	7.8	7.5	7.6
Dissolved oxygen (ppm)	6.17	5.74	5.87
Dissolved Nitrate (µgm at l ⁻¹)	23.51	21.20	22.42
Dissolved Phosphate (µgm at l ⁻¹)	1.83	1.18	1.42
Soil Organic Carbon (%)	1.16	1.84	1.6

Table 7. Water and soil quality of four selected ponds during September 2023

Parameter	P ₁	P ₂	P ₃
Surface water temperature (°C)	36.1	35.9	36.0
Surface water pH	7.8	7.5	7.6
Dissolved oxygen (ppm)	6.22	5.76	5.90
Dissolved Nitrate (µgm at l ⁻¹)	24.33	21.87	22.71
Dissolved Phosphate (µgm at l ⁻¹)	1.90	1.23	1.45
Soil Organic Carbon (%)	1.09	1.77	1.52

Table 8. Water and soil quality of four selected ponds during October 2023

Parameter	P ₁	P ₂	P ₃
Surface water temperature (°C)	35.8	35.2	35.5
Surface water pH	7.7	7.5	7.6
Dissolved oxygen (ppm)	6.30	5.72	5.99
Dissolved Nitrate (µgm at l ⁻¹)	24.73	22.21	23.10
Dissolved Phosphate (µgm at l ⁻¹)	1.96	1.30	1.51
Soil Organic Carbon (%)	0.99	1.71	1.45

The provided ANOVA tables (Tables 9 – 14) present the variations of hydrological parameters and Soil Organic Carbon (SOC) among different ponds and across different months during the culture period of *Labeo rohita* in the Moyna region, East Midnapur district, West Bengal.

Surface Water Temperature: ANOVA results indicate significant variations between months ($F = 808.2637, P < 0.0001$) and between ponds ($F = 4.384615, P = 0.037202$). This suggests that both temporal and spatial factors significantly influence the variations of surface water temperature that can affect the CI of *Labeo rohita*.

Surface Water pH: Similar to temperature, surface water pH showed significant variations between months ($F = 808.2637, P < 0.0001$) and ponds ($F = 4.384615, P = 0.037202$), indicating that the pH levels fluctuate over time and vary across different ponds.

Dissolved Oxygen (DO): The variations in dissolved oxygen were also significant between months ($F = 15.40738, P < 0.0001$) and between ponds ($F = 152.4417, P < 0.0001$), suggesting that dissolved oxygen levels are influenced by both temporal and spatial factors. The abundance of natural fish feed of carp may influence the values of DO.

Dissolved Nitrate: ANOVA results for dissolved nitrate show highly significant variations between months ($F = 42.17391, P < 0.0001$) and ponds ($F = 227.6191, P < 0.0001$), indicating that nitrate concentrations vary significantly over time and among ponds.

Dissolved Phosphate: The dissolved phosphate levels exhibited significant variations between months ($F = 226.7205, P < 0.0001$) and between ponds ($F = 6618.607, P < 0.0001$), highlighting that both temporal and spatial differences significantly affect phosphate concentrations.

Soil Organic Carbon (SOC): The SOC content showed significant variations between months ($F = 155.051, P < 0.0001$) and ponds ($F = 2549.204, P < 0.0001$), indicating that the SOC levels are influenced by both the time of the year and the specific pond.

Table 9. ANOVA for Surface Water Temperature

Source of Variation	SS	df	MS	F	P-value	F crit
Between Months	70.04952	6	11.67492	808.2637	6.31E-15	2.99612
Between Ponds	0.126667	2	0.063333	4.384615	0.037202	3.885294
Error	0.173333	12	0.014444			
Total	70.34952	20				

Table 10. ANOVA for Surface Water pH

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Months	70.04952	6	11.67492	808.2637	6.31E-15	2.99612
Between Ponds	0.126667	2	0.063333	4.384615	0.037202	3.885294
Error	0.173333	12	0.014444			
Total	70.34952	20				

Table 11. ANOVA for Dissolved Oxygen

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Months	0.271363	7	0.038766	15.40738	1.37E-05	2.764199
Between Ponds	0.767108	2	0.383554	152.4417	3.14E-10	3.738892
Error	0.035225	14	0.002516			
Total	1.073696	23				

Table 12. ANOVA for Dissolved Nitrate

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Months	12.7012	7	1.814457	42.17391	2.39E-08	2.764199
Between Ponds	19.58581	2	9.792904	227.6191	2.1E-11	3.738892
Error	0.602325	14	0.043023			
Total	32.88933	23				

Table 13. ANOVA for Dissolved Phosphate

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Months	0.216329	7	0.030904	226.7205	2.59E-13	2.764199
Between Ponds	1.804358	2	0.902179	6618.607	1.47E-21	3.738892
Error	0.001908	14	0.000136			
Total	2.022596	23				

Table 14. ANOVA for Soil Organic Carbon (SOC)

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Months	0.40571 7	7	0.05796	155.051	3.57E-12	2.764199
Between Ponds	1.90583 3	2	0.95291 7	2549.204	1.15E-18	3.738892
Error	0.00523 3	14	0.00037 4			
Total	2.31678 3	23				

These findings underscore the importance of considering both temporal and spatial variations in hydrological parameters and SOC during aquaculture practices. Future research should focus on identifying the underlying factors causing these variations to optimize aquaculture productivity and sustainability.

4. Discussion

The python program accurately determined the condition index values for the three ponds (Figs. 2 - 4), reflecting the health and well-being of the fish under different environmental conditions. Pond P₁ exhibited the highest condition index value of 1.97, indicating robust growth conditions. In contrast, Pond P₂ had the lowest condition index value of 1.35, suggesting a stressful environment of highest degree. Pond P₃ had a condition index value of 1.60, indicating better conditions than Pond P₂ but not as optimal as Pond P₁. These variations highlight the differing environmental impacts on fish health across the ponds. This study demonstrated the potential of program to monitor the condition index and management in carp aquaculture, offering a promising approach to improving productivity and sustainability in the industry. Several factors could influence the variation in condition index values across the ponds. The quality of water, including parameters such as pH, dissolved oxygen, and temperature, plays a critical role in fish health. Differences in these parameters could lead to variations in the condition index. A well-defined scientific management of water quality is needed for Pond P₂.

References

- Boyd, C. E. (1998). *Water Quality in Warmwater Fish Ponds*. Auburn University.
- Chen, C., Li, W., & Shi, Y. (2020). Application of Machine Learning in Aquaculture: A Review. *Aquaculture Research*, 51(5), 1874-1882.
- García, M., Prieto, G., & Rivas, F. (2017). Economic Benefits of Applying Machine Learning in Fish Farming. *Journal of Aquaculture Economics & Management*, 21(4), 399-416.
- Jhingran, V. G. (1991). *Fish and Fisheries of India*. Hindustan Publishing Corporation.
- Jiang, X., Huang, W., & Wu, X. (2021). Machine Learning Applications in Aquaculture: An Overview. *Aquaculture International*, 29(2), 547-561.
- Khan, R. A., & Khan, S. U. (2019). Integrating IoT and Machine Learning for Smart Aquaculture. *International Journal of Information Management*, 49(3), 123-131.
- Mishra, A., Behera, B. K., & Mohanty, B. P. (2018). Monitoring Water Quality Parameters for Sustainable Aquaculture. *Aquaculture Research*, 49(4), 1528-1540.
- Patil, P. R., & Gade, S. S. (2019). Predicting Fish Growth Using Machine Learning Techniques. *Fisheries Science*, 85(3), 439-447.

- Sharma, A., Singh, R., & Kumar, P. (2021). Classification of Fish Growth Stages Using Machine Learning Algorithms. *Aquaculture International*, 29(1), 91-105.
- Singh, V., Gupta, S., & Mehta, R. (2020). Clustering Hydrological Conditions and Fish Growth Patterns Using K-means. *Environmental Monitoring and Assessment*, 192(2), 109.
- Wang, T., Zhang, Y., & Lin, F. (2019). Real-time Monitoring and Prediction of Water Quality in Aquaculture Using IoT and Machine Learning. *Computers and Electronics in Agriculture*, 165(3), 104957.
- Zhang, X., Liu, Q., & Sun, Y. (2020). The Impact of Climate Change on Aquaculture: A Machine Learning Perspective. *Climate Change Journal*, 12(5), 387-403.