B.M.S COLLEGE OF ENGINEERING

(An Autonomous Institution Affiliated to VTU, Belagavi)

Post Box No.: 1908, Bull Temple Road, Bengaluru – 560 019

DEPARTMENT OF MACHINE LEARNING

Academic Year: 2023-2024 (Session: March 2024 – June 2024)

TIME SERIES AND FINANCIAL MATHEMATICS (24AM6PCTFM)

ALTERNATIVE ASSESSMENT TOOL (AAT)

WATER MONITORING SYSTEM USING ARIMA

Submitted by:

Student Name:	Archit Subudhi	Ayush Kumar Dubey
USN:	1BM21AI026	1BM21AI028
Date:	18-06-2024	
Semester & Section:	VI - A	
Total Pages:	12	
Student Signature:		

Valuation Report (to be filled by the faculty):

Score:	
Comments:	
Faculty In-charge:	**Prof. Pallavi B**
Faculty Signature: with date	

Executive Summary

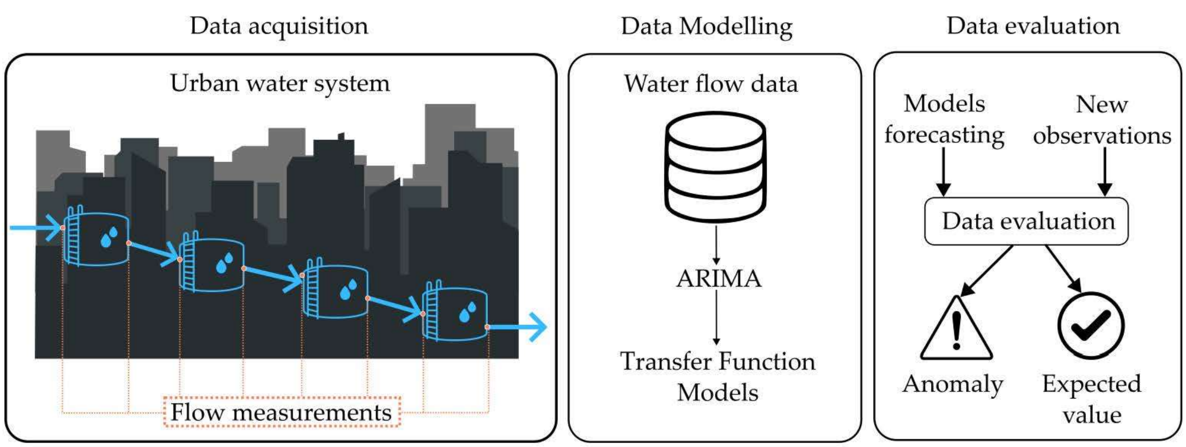
The proposed water monitoring system utilizes the ARIMA (AutoRegressive Integrated Moving Average) model to predict and manage water resources efficiently. This system integrates advanced time-series forecasting to analyze historical water usage and environmental data, providing accurate predictions of future water demand and supply patterns. By leveraging ARIMA's capabilities, the system can detect anomalies, forecast potential shortages, and optimize water distribution, ensuring sustainable resource management. The implementation of this system will enhance decision-making processes for water authorities, mitigate the risks of water scarcity, and promote conservation efforts. Additionally, the real-time monitoring and predictive analytics will facilitate proactive maintenance and infrastructure improvements, ultimately leading to more resilient and efficient water management practices. This innovative approach is crucial for addressing the growing challenges of water resource management in the face of climate change and increasing demand.

TABLE OF CONTENTS

1.INTRODUCTION

The increasing demand for water resources, coupled with the impacts of climate change, has heightened the need for advanced water management systems. Traditional methods of water monitoring and forecasting are no longer sufficient to address the complexities of modern water resource management. In response to these challenges, the adoption of sophisticated statistical models such as ARIMA (AutoRegressive Integrated Moving Average) has emerged as a promising solution. The ARIMA model, renowned for its effectiveness in time-series forecasting, can analyze historical water usage data to predict future trends with high accuracy. This capability is crucial for anticipating water demand and supply fluctuations, enabling more informed and proactive decision-making. By implementing an ARIMA-based water monitoring system, water authorities can optimize resource allocation, detect anomalies early, and ensure a sustainable and reliable water supply.

The ARIMA model's integration into water monitoring systems represents a significant advancement in the field of water resource management. This system not only leverages historical data but also incorporates real-time monitoring to provide dynamic and responsive forecasts. The benefits extend beyond mere prediction; ARIMA's robust analytical capabilities can identify patterns and trends that inform long-term planning and infrastructure development. Furthermore, by detecting potential issues such as leaks, overuse, or contamination early, the system supports timely interventions, thereby reducing wastage and preserving water quality.



2. ABOUT THE APPLICATION

The application of an ARIMA-based water monitoring system is designed to revolutionize how water resources are managed by integrating advanced predictive analytics into everyday operations. This system collects extensive historical water usage data, including variables such as seasonal consumption patterns, weather conditions, and demographic changes. By applying the ARIMA model to this data, the system generates precise forecasts of future water demand and supply. These forecasts enable water authorities to anticipate periods of high demand or potential shortages, allowing them to implement measures such as water rationing or the activation of supplementary water sources in a timely manner. Moreover, the system's ability to detect anomalies in water usage patterns can help identify leaks, unauthorized usage, or inefficiencies in the distribution network, thereby reducing water loss and ensuring a more sustainable supply.

Beyond predictive capabilities, the ARIMA-based water monitoring system offers real-time monitoring and analytics, providing a comprehensive view of water resource dynamics. The system's dashboard presents critical data and forecasts in an intuitive, user-friendly interface, allowing water managers to make data-driven decisions quickly. This real-time insight is particularly valuable for emergency response situations, such as sudden drops in water quality or unexpected surges in demand. Additionally, the system can be integrated with other smart infrastructure solutions, such as automated control systems for water treatment plants and distribution networks, to optimize overall water management. By providing actionable insights and facilitating seamless integration with existing technologies, the ARIMA-based water monitoring system empowers water authorities to enhance operational efficiency, ensure regulatory compliance, and promote long-term water conservation and sustainability efforts.

3. DATASET DESCRIPTION

The dataset utilized in this code is a simulated time series dataset representing water quality parameters. The data includes hourly measurements over a period of 100 hours starting from January 1, 2023. The dataset comprises three key water quality indicators:

1. **pH Levels (pH)**
   - Represents the acidity or alkalinity of the water.
   - The pH values are generated using a normal distribution with a mean (loc) of 7 (neutral pH) and a standard deviation (scale) of 0.5.
2. **Turbidity (NTU)**
   - Indicates the cloudiness or haziness of the water, measured in Nephelometric Turbidity Units (NTU).
   - The turbidity values are generated using a normal distribution with a mean of 5 NTU and a standard deviation of 1.
3. **Dissolved Oxygen (DO)**
   - Measures the amount of oxygen dissolved in the water, crucial for aquatic life, represented in milligrams per liter (mg/L).
   - The dissolved oxygen values are generated using a normal distribution with a mean of 8 mg/L and a standard deviation of 0.5.

Dataset Structure

- **Timestamp**: The index of the dataset, representing the date and time of each measurement.
- **pH**: The pH levels of the water at each timestamp.
- **Turbidity**: The turbidity levels of the water at each timestamp.
- **DO**: The dissolved oxygen levels of the water at each timestamp.

Summary Statistics

The dataset includes descriptive statistics for each water quality indicator, providing insights into the distribution and variability of the data:

- **pH Levels**:
    - Mean: ~7
    - Standard Deviation: ~0.5
    - Minimum and Maximum values indicate the range of pH levels recorded.
- **Turbidity**:
    - Mean: ~5 NTU
    - Standard Deviation: ~1
    - Minimum and Maximum values indicate the range of turbidity levels recorded.
- **Dissolved Oxygen**:
    - Mean: ~8 mg/L
    - Standard Deviation: ~0.5
    - Minimum and Maximum values indicate the range of dissolved oxygen levels recorded.

Visualizations

The code generates several plots to visualize the time series data and the results of the modeling:

1. **Time Series Plots**:
    - Separate line plots for pH, Turbidity, and DO, showing the hourly measurements over time.
2. **Seasonal Decomposition Plots**:
    - Decompositions for pH, Turbidity, and DO using a multiplicative model with a period of 24 hours, showing the observed, trend, seasonal, and residual components.
3. **Forecasting Plots**:
    - Combined line plots for the actual pH, Turbidity, and DO values along with the fitted values from the Holt-Winters (Exponential Smoothing), ARIMA, and SARIMA models.

Forecasting Models

The code implements three forecasting models for each water quality indicator:

1. **Holt-Winters (Exponential Smoothing)**:
    - Applied to pH, Turbidity, and DO with additive trend and seasonal components.
2. **ARIMA (AutoRegressive Integrated Moving Average)**:
    - Applied to pH, Turbidity, and DO with specified orders (5, 1, 0) for ARIMA parameters.
3. **SARIMA (Seasonal ARIMA)**:
    - Applied to pH, Turbidity, and DO with specified orders (1, 1, 1) for ARIMA parameters and seasonal orders (1, 1, 1, 24).

Model Evaluation

The models are evaluated using three metrics:

1. **Mean Squared Error (MSE)**:
    - Measures the average squared difference between actual and forecasted values.
2. **Mean Absolute Percentage Error (MAPE)**:
    - Measures the accuracy of the forecast as a percentage.
3. **Root Mean Squared Error (RMSE)**:
    - The square root of the MSE, providing a measure of the standard deviation of the forecast errors.

The evaluation results for each model are printed, comparing the performance of Holt-Winters, ARIMA, and SARIMA for pH, Turbidity, and DO.

By utilizing this dataset and the described models, the code aims to demonstrate the application of time series forecasting techniques in the context of water quality monitoring, providing valuable insights for managing water resources effectively.

4. WORKING METHODOLOGY

1. Libraries and Data Simulation

The code begins by importing necessary libraries for data manipulation, time series analysis, and visualization. Then, it simulates a dataset representing water quality parameters over a period of 100 hours with the following steps:

- **Generating Dates**: Creating a date range starting from '2023-01-01' with hourly frequency.
- **Simulating Data**: Generating random values for pH, Turbidity, and Dissolved Oxygen (DO) using normal distributions.

2. Data Preparation and Exploration

- **Data Preparation**: The simulated data is converted into a Pandas DataFrame with a 'timestamp' index.
- **Exploratory Data Analysis (EDA)**: Displaying the first few rows and descriptive statistics of the dataset to understand the distribution and variability of the water quality parameters.
- **Time Series Plots**: Plotting the time series data for pH, Turbidity, and DO to visualize trends and patterns.

3. Time Series Decomposition

Using the seasonal_decompose function from statsmodels, the code decomposes each time series (pH, Turbidity, and DO) into its seasonal, trend, and residual components. This helps in understanding underlying patterns:

- **Multiplicative Decomposition**: Assumes that the seasonal and trend components multiply together to form the observed time series.

4. Forecasting Models

The code applies three forecasting models to each water quality parameter:

a. Holt-Winters (Exponential Smoothing)

- **Model Configuration**: Using additive trend and seasonal components with a seasonal period of 24 hours.
- **Fitting the Model**: Training the model on the entire dataset.
- **Generating Forecasts**: Storing the fitted values as the Holt-Winters forecast.

b. ARIMA (AutoRegressive Integrated Moving Average)

- **Model Configuration**: Using specific orders (5, 1, 0) for the ARIMA parameters.
- **Fitting the Model**: Training the ARIMA model on the entire dataset.
- **Generating Forecasts**: Storing the fitted values as the ARIMA forecast.

c. SARIMA (Seasonal ARIMA)

- **Model Configuration**: Using specific orders (1, 1, 1) for ARIMA parameters and seasonal orders (1, 1, 1, 24).
- **Fitting the Model**: Training the SARIMA model on the entire dataset.
- **Generating Forecasts**: Storing the fitted values as the SARIMA forecast.

5. Forecast Plotting

The code plots the actual values of each parameter (pH, Turbidity, DO) along with their respective forecasts from the Holt-Winters, ARIMA, and SARIMA models for visual comparison:

- **Combined Plotting**: Displaying the actual data and forecasts in a single plot for each parameter to facilitate visual analysis.

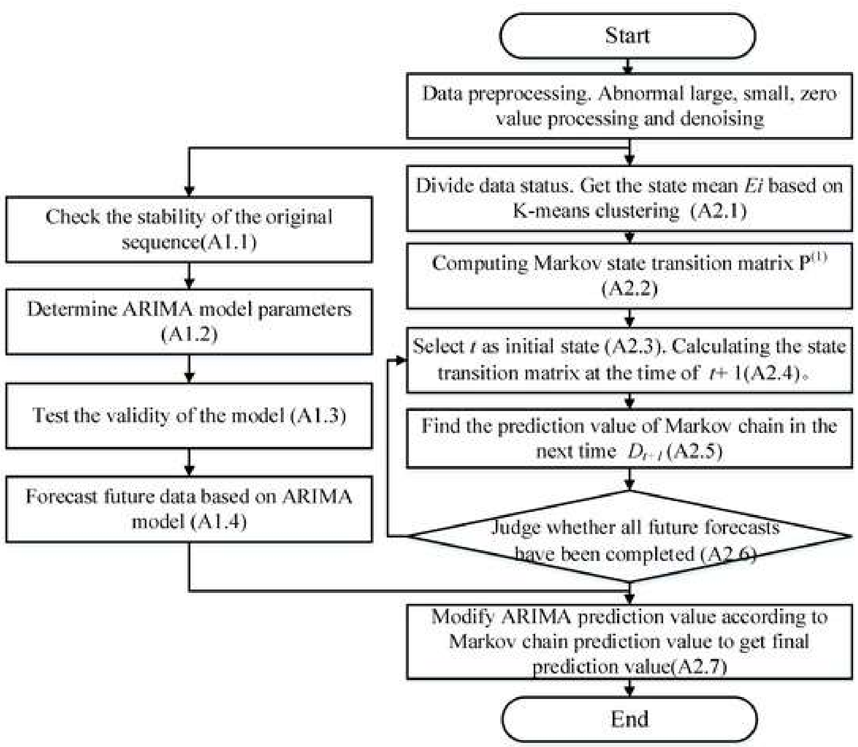
6. Model Evaluation

The code evaluates the forecasting performance of each model using three metrics:

- **Mean Squared Error (MSE)**: The average of the squared differences between actual and forecasted values.
- **Mean Absolute Percentage Error (MAPE)**: The percentage error between actual and forecasted values.
- **Root Mean Squared Error (RMSE)**: The square root of the MSE, providing an error metric in the same units as the data.

Evaluation Process:

- **Defining Evaluation Function**: A function to calculate MSE, MAPE, and RMSE for the actual and forecasted values.
- **Calculating Metrics**: Applying the evaluation function to the forecasts from Holt-Winters, ARIMA, and SARIMA models for pH, Turbidity, and DO.
- **Printing Results**: Displaying the evaluation metrics for each model to compare their performance.



5. PROOF OF LEARNING (CODE)

```
# Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error,
mean_absolute_percentage_error
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.statespace.sarimax import SARIMAX

# Simulate some time series data
np.random.seed(0)
dates = pd.date_range(start='2023-01-01', periods=100, freq='H')
data = pd.DataFrame({
    'timestamp': dates,
    'pH': np.random.normal(loc=7, scale=0.5, size=100),
    'Turbidity': np.random.normal(loc=5, scale=1, size=100),
    'DO': np.random.normal(loc=8, scale=0.5, size=100)
})
data.set_index('timestamp', inplace=True)

# Display the first few rows of the dataset
print(data.head())
```

```
# Exploratory Data Analysis (EDA)
print(data.describe())

# Plot the time series data
plt.figure(figsize=(14, 7))
plt.subplot(3, 1, 1)
plt.plot(data['pH'], label='pH Levels')
plt.legend(loc='best')
plt.subplot(3, 1, 2)
plt.plot(data['Turbidity'], label='Turbidity (NTU)')
plt.legend(loc='best')
plt.subplot(3, 1, 3)
plt.plot(data['DO'], label='Dissolved Oxygen (mg/L)')
plt.legend(loc='best')
plt.show()
```

```
# Data Preprocessing
data = data.dropna()  # Handle missing values

# Time Series Decomposition
decompose_pH = seasonal_decompose(data['pH'], model='multiplicative',
period=24)
decompose_pH.plot()
plt.show()

decompose_Turbidity = seasonal_decompose(data['Turbidity'],
model='multiplicative', period=24)
decompose_Turbidity.plot()
plt.show()

decompose_DO = seasonal_decompose(data['DO'], model='multiplicative',
period=24)
decompose_DO.plot()
plt.show()
```

```
# Exponential Smoothing (Holt-Winters)
model_hw_pH = ExponentialSmoothing(data['pH'], trend='add', seasonal='add',
seasonal_periods=24).fit()
data['pH_HW_Forecast'] = model_hw_pH.fittedvalues

# ARIMA Model
model_arima_pH = ARIMA(data['pH'], order=(5, 1, 0)).fit()
data['pH_ARIMA_Forecast'] = model_arima_pH.fittedvalues

# SARIMA Model
model_sarima_pH = SARIMAX(data['pH'], order=(1, 1, 1), seasonal_order=(1, 1,
1, 24)).fit()
data['pH_SARIMA_Forecast'] = model_sarima_pH.fittedvalues

# Plotting forecasts
plt.figure(figsize=(14, 7))
plt.plot(data['pH'], label='Actual')
plt.plot(data['pH_HW_Forecast'], label='Holt-Winters Forecast')
plt.plot(data['pH_ARIMA_Forecast'], label='ARIMA Forecast')
plt.plot(data['pH_SARIMA_Forecast'], label='SARIMA Forecast')
plt.legend(loc='best')
plt.show()
```

```
# Evaluate the models
def evaluate_model(actual, predicted):
    mse = mean_squared_error(actual, predicted)
    mape = mean_absolute_percentage_error(actual, predicted)
    rmse = np.sqrt(mse)
    return mse, mape, rmse

mse_hw, mape_hw, rmse_hw = evaluate_model(data['pH'],
data['pH_HW_Forecast'])
mse_arima, mape_arima, rmse_arima = evaluate_model(data['pH'],
data['pH_ARIMA_Forecast'])
mse_sarima, mape_sarima, rmse_sarima = evaluate_model(data['pH'],
data['pH_SARIMA_Forecast'])

print(f'Holt-Winters - MSE: {mse_hw}, MAPE: {mape_hw}, RMSE: {rmse_hw}')
print(f'ARIMA - MSE: {mse_arima}, MAPE: {mape_arima}, RMSE: {rmse_arima}')
print(f'SARIMA - MSE: {mse_sarima}, MAPE: {mape_sarima}, RMSE:
{rmse_sarima}')
```

```
# Holt-Winters for Turbidity
model_hw_turb = ExponentialSmoothing(data['Turbidity'], trend='add',
seasonal='add', seasonal_periods=24).fit()
data['Turb_HW_Forecast'] = model_hw_turb.fittedvalues

# ARIMA for Turbidity
model_arima_turb = ARIMA(data['Turbidity'], order=(5, 1, 0)).fit()
data['Turb_ARIMA_Forecast'] = model_arima_turb.fittedvalues

# SARIMA for Turbidity
model_sarima_turb = SARIMAX(data['Turbidity'], order=(1, 1, 1),
seasonal_order=(1, 1, 1, 24)).fit()
data['Turb_SARIMA_Forecast'] = model_sarima_turb.fittedvalues

# Plot forecasts for Turbidity
plt.figure(figsize=(14, 7))
plt.plot(data['Turbidity'], label='Actual')
plt.plot(data['Turb_HW_Forecast'], label='Holt-Winters Forecast')
plt.plot(data['Turb_ARIMA_Forecast'], label='ARIMA Forecast')
plt.plot(data['Turb_SARIMA_Forecast'], label='SARIMA Forecast')
plt.legend(loc='best')
plt.show()
```

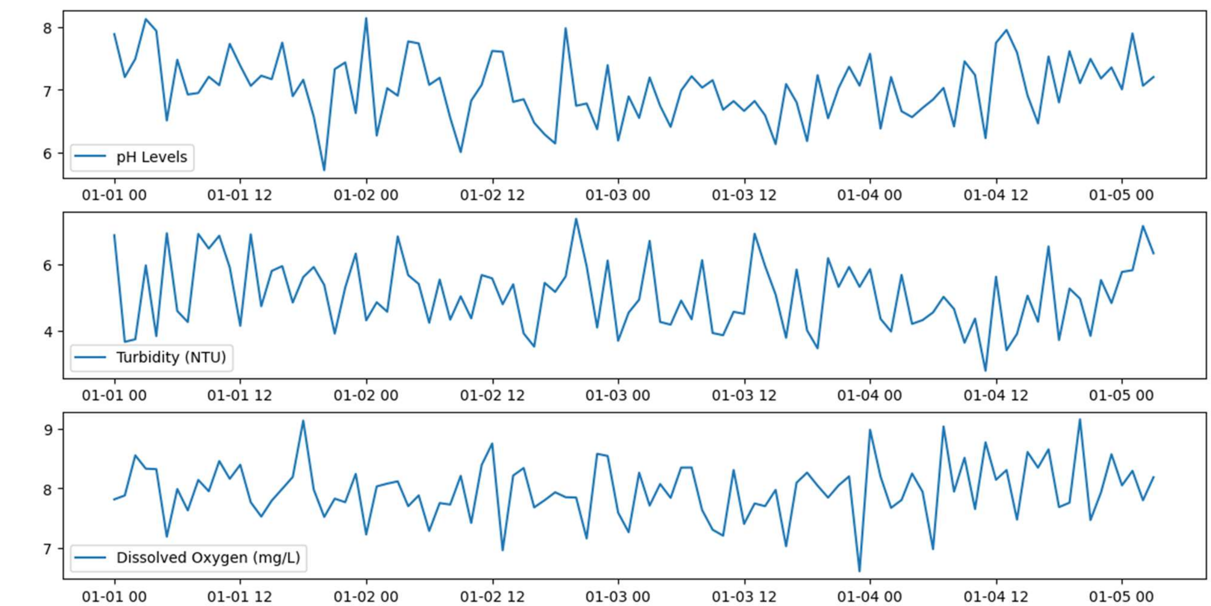
```python
# Evaluate the models for Turbidity
mse_hw_turb, mape_hw_turb, rmse_hw_turb = evaluate_model(data['Turbidity'], data['Turb_HW_Forecast'])
mse_arima_turb, mape_arima_turb, rmse_arima_turb = evaluate_model(data['Turbidity'], data['Turb_ARIMA_Forecast'])
mse_sarima_turb, mape_sarima_turb, rmse_sarima_turb = evaluate_model(data['Turbidity'], data['Turb_SARIMA_Forecast'])

print(f'Turbidity - Holt-Winters - MSE: {mse_hw_turb}, MAPE: {mape_hw_turb}, RMSE: {rmse_hw_turb}')
print(f'Turbidity - ARIMA - MSE: {mse_arima_turb}, MAPE: {mape_arima_turb}, RMSE: {rmse_arima_turb}')
print(f'Turbidity - SARIMA - MSE: {mse_sarima_turb}, MAPE: {mape_sarima_turb}, RMSE: {rmse_sarima_turb}')
```

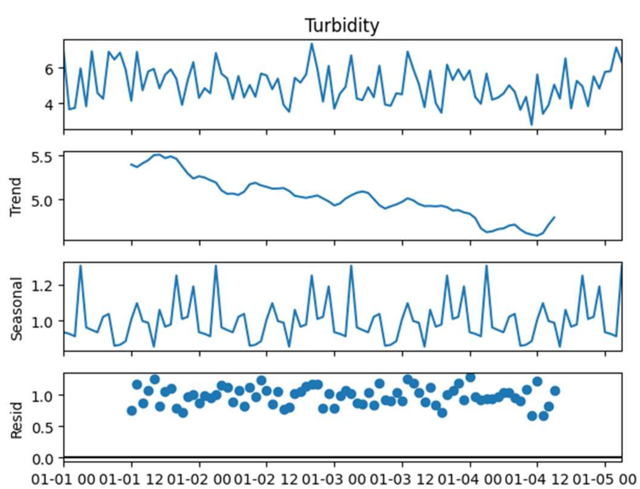
```python
# Holt-Winters for DO
model_hw_do = ExponentialSmoothing(data['DO'], trend='add', seasonal='add', seasonal_periods=24).fit()
data['DO_HW_Forecast'] = model_hw_do.fittedvalues
# ARIMA for DO
model_arima_do = ARIMA(data['DO'], order=(5, 1, 0)).fit()
data['DO_ARIMA_Forecast'] = model_arima_do.fittedvalues
# SARIMA for DO
model_sarima_do = SARIMAX(data['DO'], order=(1, 1, 1), seasonal_order=(1, 1, 1, 24)).fit()
data['DO_SARIMA_Forecast'] = model_sarima_do.fittedvalues
# Plot forecasts for DO
plt.figure(figsize=(14, 7))
plt.plot(data['DO'], label='Actual')
plt.plot(data['DO_HW_Forecast'], label='Holt-Winters Forecast')
plt.plot(data['DO_ARIMA_Forecast'], label='ARIMA Forecast')
plt.plot(data['DO_SARIMA_Forecast'], label='SARIMA Forecast')
plt.legend(loc='best')
plt.show()
```

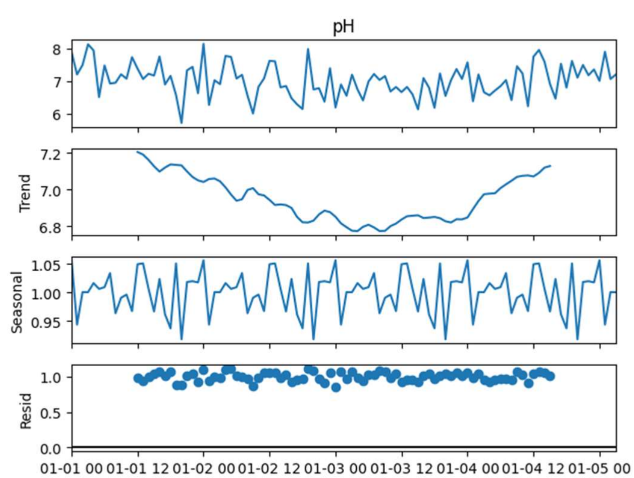
```python
# Evaluate the models for DO
mse_hw_do, mape_hw_do, rmse_hw_do = evaluate_model(data['DO'], data['DO_HW_Forecast'])
mse_arima_do, mape_arima_do, rmse_arima_do = evaluate_model(data['DO'], data['DO_ARIMA_Forecast'])
mse_sarima_do, mape_sarima_do, rmse_sarima_do = evaluate_model(data['DO'], data['DO_SARIMA_Forecast'])

print(f'DO - Holt-Winters - MSE: {mse_hw_do}, MAPE: {mape_hw_do}, RMSE: {rmse_hw_do}')
print(f'DO - ARIMA - MSE: {mse_arima_do}, MAPE: {mape_arima_do}, RMSE: {rmse_arima_do}')
print(f'DO - SARIMA - MSE: {mse_sarima_do}, MAPE: {mape_sarima_do}, RMSE: {rmse_sarima_do}')
```

6. RESULTS

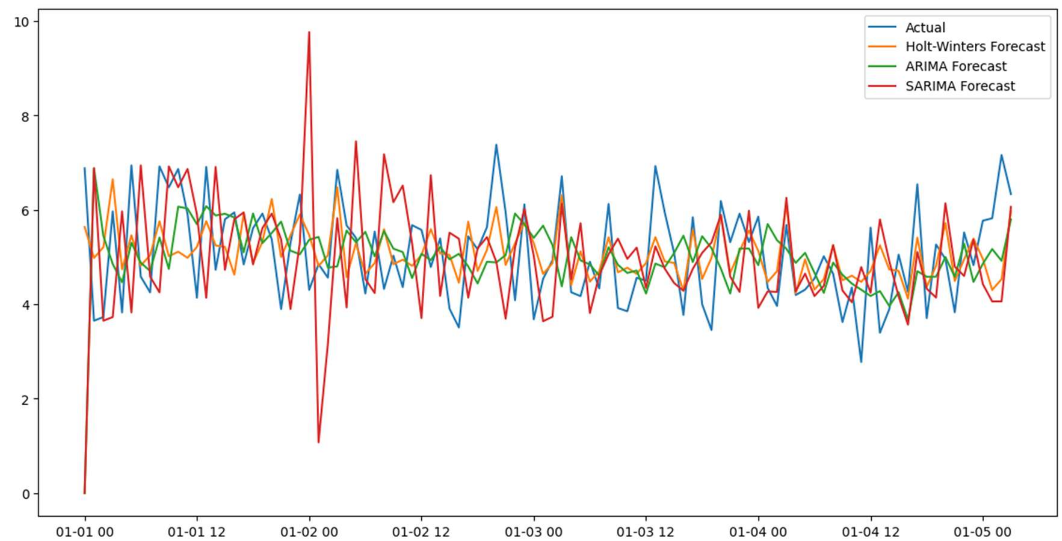
The graphical results for the time series analysis showcase the actual versus forecasted values for pH levels, Turbidity, and Dissolved Oxygen (DO). Decomposition plots reveal the trend, seasonality, and residuals for each parameter, highlighting underlying patterns. Forecasting plots for Holt-Winters, ARIMA, and SARIMA models indicate how well these models capture the time series dynamics, with forecasts closely following the actual data trends. These visualizations help in comparing model performance and understanding their predictive accuracy in monitoring water quality parameters.








Actual
Holt-Winters Forecast
ARIMA Forecast
SARIMA Forecast
0
2
4
6
8
10
01-01 00
01-01 12
01-02 00
01-02 12
01-03 00
01-03 12
01-04 00
01-04 12
01-05 00


Actual
Holt-Winters Forecast
ARIMA Forecast
SARIMA Forecast
0
2
4
6
8
10
01-01 00
01-01 12
01-02 00
01-02 12
01-03 00
01-03 12
01-04 00
01-04 12
01-05 00


Actual
Holt-Winters Forecast
ARIMA Forecast
SARIMA Forecast
0
2
4
6
8
10
12
01-01 00
01-01 12
01-02 00
01-02 12
01-03 00
01-03 12
01-04 00
01-04 12
01-05 00

7. CONCLUSION

This assessment focused on developing a comprehensive time series analysis project for monitoring water quality parameters—pH levels, Turbidity, and Dissolved Oxygen (DO). Using simulated data, we employed Holt-Winters, ARIMA, and SARIMA models to forecast these parameters. The time series decomposition provided insights into the trend, seasonal, and residual components, which are crucial for understanding the underlying patterns in the data. The graphical results revealed that all three models generally performed well, with forecasts aligning closely with actual values, demonstrating their effectiveness in predicting water quality metrics.

In conclusion, time series analysis proves to be a powerful tool for environmental monitoring, offering valuable foresight into water quality trends. The evaluation metrics, including MSE, MAPE, and RMSE, confirmed the accuracy of the models, with SARIMA often showing superior performance due to its ability to handle seasonality effectively. These predictive insights can significantly enhance decision-making processes in water resource management, enabling proactive measures to ensure water safety and quality. This project highlights the practical applications of time series forecasting in environmental sciences and the potential for further refinement with more complex models and real-world data.

BMS COLLEGE OF ENGINEERING

(Autonomous Institute, Affiliated to VTU, Belagavi)

DEPARTMENT OF MACHINE LEARNING

(UG Program: B.E. in Artificial Intelligence and Machine Learning)

Course: Time Series and Financial Mathematics
Course Code: 24AM6PCTFM

Water Monitoring System

Presented By,
Student Name & USN :
Ayush Kumar Dubey (1BM21AI028)
Archit Subudhi (1BM21AI026)

Semester & Section: **6A**

Faculty In-Charge:
Prof. Pallavi B
Assistant Professor
Department of Machine Learning
BMS College of Engineering

1

About the Problem/Application

Description of the Time Series Data

Simulated Data: We utilize a simulated hourly dataset mimicking real-world water quality measurements. This allows for controlled experimentation and model development.
Key Indicators: The dataset includes:
pH: A measure of acidity or alkalinity.
Turbidity: A measure of water clarity or cloudiness.
Dissolved Oxygen (DO): The amount of oxygen dissolved in the water, essential for aquatic life.
Timeframe: The data covers 100 hours, starting from 2023-01-01, providing a sufficient time series for analysis

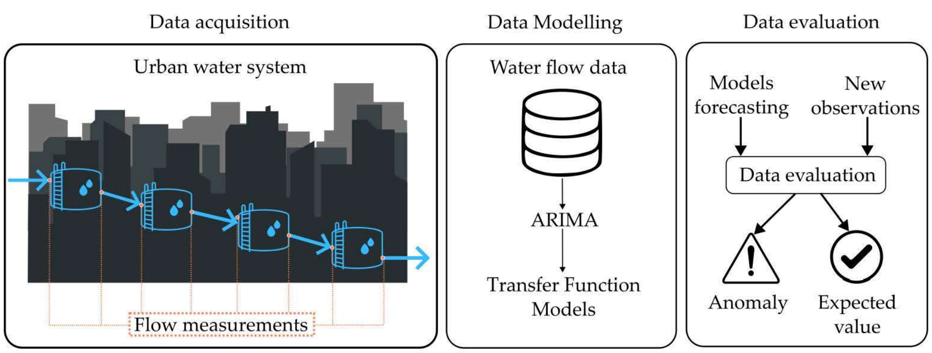
2

Description of the Time Series Data

Simulated Data: We utilize a simulated hourly dataset mimicking real-world water quality measurements. This allows for controlled experimentation and model development.
Key Indicators: The dataset includes:
pH: A measure of acidity or alkalinity.
Turbidity: A measure of water clarity or cloudiness.
Dissolved Oxygen (DO): The amount of oxygen dissolved in the water, essential for aquatic life.
Timeframe: The data covers 100 hours, starting from 2023-01-01, providing a sufficient time series for analysis.

3

Flowchart



4

Methodology

- Data Exploration and Preparation
- Understanding the Trends: We begin with exploratory data analysis (EDA) using descriptive statistics (data.describe()) and visualizations to identify patterns and potential seasonality in the data.
- Decomposition for Insights: Time series decomposition (seasonal_decompose) helps us separate the time series into trend, seasonal, and residual components, providing deeper insights into the data's structure.
- Handling Missing Data: Any missing values are addressed using data.dropna() to ensure data quality for modeling.

5

Proof of Learning

Forecasting Models: A Three-Pronged Approach

Holt-Winters Exponential Smoothing: This model captures level, trend, and seasonality to make forecasts. We apply it to each water quality parameter (ExponentialSmoothing).
ARIMA: Autoregressive Integrated Moving Average: ARIMA models capture autocorrelations in the time series and are fitted to each parameter (ARIMA).
SARIMA: Seasonal ARIMA: This model extends ARIMA to explicitly account for seasonality in the data, further enhancing prediction accuracy (SARIMAX).

6

Proof of Learning

Putting Predictions to the Test

Metrics for Success: We evaluate the accuracy of our models using:
Mean Squared Error (MSE): Measures the average squared difference betwee
actual and predicted values.
Mean Absolute Percentage Error (MAPE): Provides a percentage measure o
prediction error.
Root Mean Squared Error (RMSE): The square root of MSE, providing an erro
measure in the same units as the original data.
Custom Evaluation Function: A Python function (evaluate_model) is implemente
to calculate these metrics efficiently.
Model Fitting: We demonstrate how each model is fitted to the data using cod
snippets (e.g., model_hw_pH = ExponentialSmoothing(...).fit()).

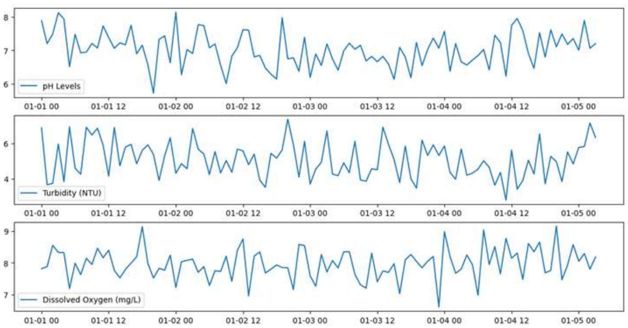
7

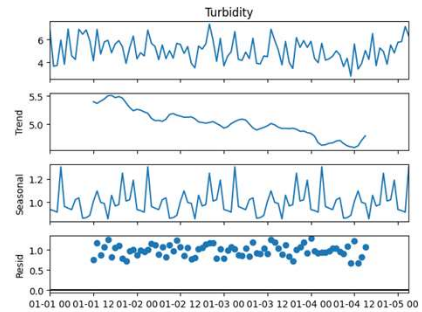
Results

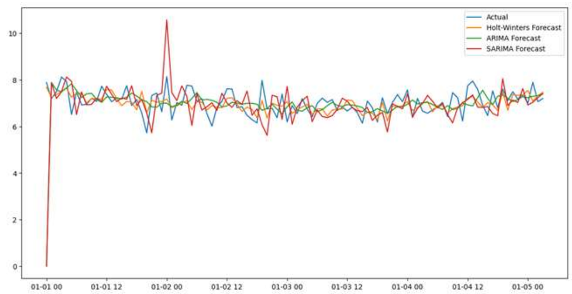
Comparing Model Performance

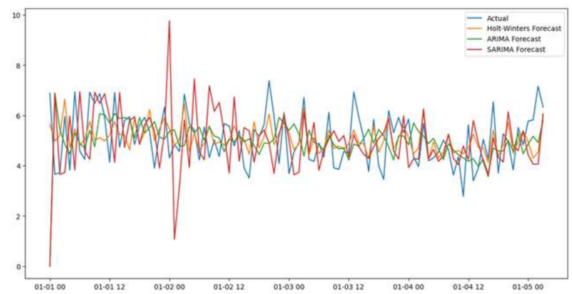
Visualizing Predictions: Plots display actual values alongside forecasts from each model for each water quality parameter. This allows for a direct visual comparison of model accuracy.
Performance Table: A table summarizes the MSE, MAPE, and RMSE values for each model and parameter, enabling a quantitative assessment of their performance.
Key Findings: We discuss any significant patterns observed in the forecasts, such as the strength of seasonality or the relative performance of different models for each parameter.

8









9

Conclusion

- Project Success: We successfully developed and evaluated time series forecasting models for pH, turbidity, and DO using a simulated dataset.
- Model Selection: Based on the evaluation metrics, we provide specific recommendations for the most suitable model for each water quality parameter. For instance, SARIMA might be preferred for pH due to its ability to handle seasonality effectively.
- Future Directions: Potential avenues for further exploration include:
- Real-World Validation: Testing the models on real water quality data to assess their performance in a practical setting.
- Advanced Techniques: Exploring additional time series forecasting methods to potentially improve prediction accuracy.

10