



Republic of Iraq

Ministry of Higher Education & Scientific Research

University of Kerbala

College of Engineering

Civil Engineering Department

**MODELLING AND PREDICTING OF RESIDUAL CHLORINE
FOR DRINKING WATER USING AI IN KARBALA CITY**

A Thesis Submitted to the Council of the Faculty of the College of the
Engineering/University Of Kerbala in Partial Fulfillment of the
Requirements for the Master Degree in Civil Engineering

By:

Nidaa Habeeb Ibrahim

Supervisors

Prof.Dr.Saif Salah Mahdi Alquzweeni

Asst. Prof. Dr. Hayder Hussein Alwan Algretawee

June 2025

Dhu Al-Hijjah 1446



Republic of Iraq

Ministry of Higher Education & Scientific Research

University of Kerbala

College of Engineering

Civil Engineering Department

**MODELLING AND PREDICTING OF RESIDUAL CHLORINE
FOR DRINKING WATER USING AI IN KARBALA CITY**

A Thesis Submitted to the Council of the Faculty of the College of the
Engineering/University Of Kerbala in Partial Fulfillment of the
Requirements for the Master Degree in Civil Engineering

By:

Nidaa Habeeb Ibrahim

Supervisors

Prof.Dr.Saif Salah Mahdi Alquzweeni

Asst. Prof. Dr. Hayder Hussein Alwan Algretawee

June 2025

Dhu Al-Hijjah 1446

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

نَرْفَعُ دَرَجَاتٍ مَّنْ نَّشَاءُ ۖ وَفَوْقَ كُلِّ ذِي عِلْمٍ عَلِيمٌ

(يوسف: جزء من الآية 76)

Examination committee certification

We certify that we have read the thesis entitled "**MODELLING AND PREDICTING OF RESIDUAL CHLORINE FOR DRINKING WATER USING AI IN KARBALA CITY** " and as an examining committee, we examined the student "**Nidaa Habeeb Ibrahim**" in its content and in what is connected with it and that, in our opinion, it is adequate as a thesis for the degree of Master of Science in Civil Engineering.

Supervisor

Signature:

Name : Prof.Dr.Saif Salah Alquzweeni

Date: / / 2025

Supervisor

Signature:

Name : Asst. Prof. Dr. Hayder Hussein Alwan Algretawee

Date: / / 2025

Member

Signature:

Name : Asst.Prof. Dr. Muhammad A. Abdulredha

Date: / / 2025

Member

Signature:

Name : Asst.prof/Dr.Sumayah Amal al-din Majeed

Date: 22/9 / 2025

Chairman

Signature:

Name: Asst.prof.Dr.Ali Jalil Chabuk

Date: / / 2025

Signature:

Name : Asst .Prof.Dr. Aysar Tuama Al-Awadi
Head of the Department of Civil Engineering

Date: / / 2025

Signature:

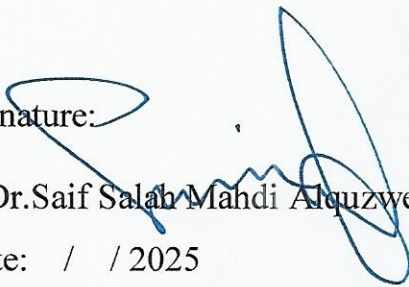
Name : Prof. Dr. Haider Nadhom Azziz

Dean of the Engineering College

Date: / / 2025

Supervisor certificate

We certify that the thesis entitled **"MODELLING AND PREDICTING OF RESIDUAL CHLORINE FOR DRINKING WATER USING AI IN KARBALA CITY "** was prepared by **Nidaa Habeeb Ibrahim** under our supervision at the Department of Civil Engineering, Faculty of Engineering, University of Kerbala as a partial of fulfilment of the requirements for the Degree of Master of Science in Civil Engineering.

Signature: 

Prof.Dr.Saif Salah Mandi Alqazweeni

Date: / / 2025

Signature: 

Asst. Prof. Dr. Hayder Hussein Alwan Alqretawee

Date: / / 2025

Linguistic certificate

I certify that the thesis entitled "**MODELLING AND PREDICTING OF RESIDUAL CHLORINE FOR DRINKING WATER USING AI IN KARBALA CITY**," which has been submitted by **Nidaa Habeeb Ibrahim**, has been proofread, and its language has been amended to meet the English style.



Signature:

Dr. Awad Ali Sagheer

Date: / / 2025

Undertaking

I certify that research work titled " **MODELLING AND PREDICTING OF RESIDUAL CHLORINE FOR DRINKING WATER USING AI IN KARBALA CITY** " is my own work. The work has not been presented elsewhere for assessment. Where material has been used from other sources, it has been properly acknowledged / referred.

Signature:

Nidaa Habeeb Ibrahim

Date: / / 2025

Dedication

لأن ما كان لله ... باقي

نذرت الطريق، وعثرات الطريق، لوجهه الكريم

Acknowledgments

Thanks be to God, first and foremost, whose guidance has always preceded my steps and who has been my support when certainty waned and I thought I was alone.

I am deeply grateful to my dear father and beloved mother, for their prayers, reassurance, and unconditional support, and for their patience with my fluctuations, as if they were an endless homeland.

To my sisters and brother, you are the friends who resemble no one else.

And to the friends whose presence made a difference.

I extend my sincere thanks to my esteemed thesis supervisors, Dr. Saif and Dr. Haidar.

I would like to thank Engineer Muayad Jaloukhani, head of the GIS department at the Karbala Water Directorate, as well as Mr. Hussein Awad at the Karbala Water Laboratory.

And to all the staff of the Public Health Laboratory, the Environment Directorate Laboratory in Karbala, and the Karbala University Laboratory.

To everyone who has brightened my mind or helped me even with just a word... Thank you.

With appreciation and gratitude.

Abstract

Ensuring the safety of drinking water is vital for public health, with effective disinfection playing a central role. Chlorination is widely applied, as it provides a residual protective effect within the distribution network. However, maintaining an optimal residual level is critical: low values reduce disinfection efficiency, while excessive levels accelerate pipe corrosion and pose health risks. This study investigated water quality at the central water plant in Karbala (Al-Hussein neighborhood), where chlorine is used for disinfection. To address this, thirty AI models were trained using five algorithms, and Compliance with the World Health Organization (2017) guidelines and the Iraqi drinking water standard No. 417 (2009) was assessed. Physical and chemical analyses were conducted at the plant and along two main pipelines. Results showed that most parameters were within acceptable limits, except turbidity and residual chlorine, related to insufficient turbidity treatment at the plant. Such increases might negatively affect the condition of the pipes within the network and inhibit the efficacy of the disinfectant. As for the residual chlorine percentage, it has consistently exceeded the permissible limits at almost all points. This offers a cumulative harm to customers while causing damage to the pipes through its interaction with the pipe walls. And these results validate the necessity for the plant to set up a monitoring system that can control and make proactive decisions about chlorine levels at specified monitoring locations. In terms of modeling, the algorithms that were used are : Random Forest (RF), Extreme Gradient Boosting (XGBoost), General Regression Neural Network (GRNN), Multi Layer Perceptron (MLP), and Transformer. By training each algorithm on three different scenarios by using Python based on the inputs.

Because the R^2 values were almost similar and high in all scenarios, the Mean Squared Error (MSE) was employed as the primary metric for the final evaluation and comparison of the models. The study results demonstrated that the Transformer achieved MSE values ranging from 0.053 to 0.101 across both directions and scenarios, while the GRNN model exhibited a range of 0.009 to 0.219. Both models demonstrated excellent performance and closely competed for the first rank. The second rank was attained by the XGBoost model with an MSE range of 0.102 to 0.335, followed by the RF with values between 0.197 and 0.595. The MLP ranked last, recording the highest error values (0.117–0.625), reflecting the weakest and least stable performance across the different scenarios. This study concluded that the process of water purification quality in the plant needs to improve, and smart AI models can be applied as a supportive tool in the proactive prediction of residual chlorine to avoid any human errors during the application of chlorine dosage.

Keywords:

Residual Chlorine, Drinking Water ,Disinfection , GRNN ,XGBoosr , Random forest , MLP, Transformer, AI, ML, ANN

Table of Contents

Examination committee certification	ii
Supervisor certificate	iii
Linguistic certificate.....	iv
Undertaking.....	v
Dedication	vi
Acknowledgments.....	vii
Abstract	viii
Table of Contents	x
List of Figures	xv
List of Symbols	xxi
Chapter One: INTRODUCTION	1
1.1 Overview	1
1.2 Research Goals.....	3
1.3 Research Objectives	4
1.4 Thesis structure	4
Chapter Two: LITERATURE REVIEW	6
2.1 Introduction.....	6
2.2 General	6
2.3 Water treatment plants	6
2.3.1 Coagulation	7
2.3.2 Flocculation	7
2.3.3 Sedimentation	7
2.3.4 Filtration.....	8
2.3.5 Disinfection.....	8
2.3.5.1 Disinfection with chlorine	8
2.3.5.1.1 Chlorine dioxide.....	10

2.3.5.1.2	Chlorine chemistry	11
2.3.5.1.3	Chlorine Zones	12
2.3.5.2	Disinfection by ozone	13
2.3.5.3	Disinfection by UV radiation	14
2.3.6	Comparison of disinfecting techniques	15
2.4.1	Random Forest (RF)	17
2.4.2	General Regression Neural Network (GRNN)	18
2.4.3	Multi-Layer Perceptron (MLP)	20
2.3.6	Extreme Gradient Boosting (XGBoost) algorithm.....	21
2.4.5	Transformer Model	22
2.5	Previous Study	23
2.6	Summary	28
Chapter Three: Methodology.....		30
3.1	Introduction	30
3.2	Research region	32
3.3.3.1	Free RC test.....	36
3.3.3.2	PH test.....	37
3.3.3.3	Electrical conductivity (EC) and Dissolved solids testing (TDS) and Temperature (Temp) test.....	38
3.3.3.4	Turbidity test	39
3.3.3.5	Chloride test (Cl^-)	40
3.3.3.6	Sulfate test (SO_4^{-2})	41
3.3.3.7	Total Hardness test (TH).....	42
3.3.3.8	Calcium test (Ca^{+2})	43
3.3.3.9	Alkalinity test (Alk).....	44
3.4	Programming Work.....	45
3.4.1	Cleaning data.....	45

3.4.2	The software used.....	46
3.4.2.1	Python 3	46
3.4.2.2	Jupyter Notebook	47
3.4.2.3	Anaconda	47
3.4.2.4	SPSS.....	48
Chapter Four: Results and Discussion		59
4.1	Introduction	59
4.2	Data Analysis	59
4.2.1	PH	59
4.2.2	Turbidity	60
4.2.3	Residual Chlorine	63
4.2.4	Total Dissolved Solid (TDS)	66
4.2.5	Total Hardness (TH)	67
4.2.6	Calcium (Ca^{+2})	67
4.2.7	Chloride (Cl^-).....	68
4.2.8	Sulfate (SO_4^{-2}).....	68
4.2.9	Temperature	69
4.2.10	Electrical conductivity	69
4.2.11	Alkalinity	70
4.3	Data Augmentation	71
4.4	The model outcomes	74
4.4.1	Random Forest Model.....	74
4.4.2	XGBoost Regression model.....	81
4.4.4	MLP Regression.....	87
4.4.5	GRNN Mode.....	96
4.4.6	Transformers Model.....	105
4.5	Comparison of the models performance	110

Chapter Five: Conclusions and Recommendations	118
5.1 Introduction	118
5.2 Conclusion	118
5.3 Recommendations	119
REFERENCES.....	1
Appendices	11

List of Tables

Table (4-1) Original data for direct one	71
Table (4-2) Augmentation data for direct one	72
Table (4-3): t-test and P value result s for direct 1.	72
Table (4-4): Original data direct 2.	73
Table (4-5): Augmentation direct 2.	74
Table (4-6): t-test and P value result direct 2.....	74
Table (4-7): the hyperparameters for RF direct 1.	77
Table (4-8): Performance metric for RF direct 1.	77
Table (4-9): the hyperparameters for RF direct 2	79
Table (4-10): Performance metric for RF direct 2.....	79
Table (4-11): the hyper parameters for XGBoost direct 1.....	83
Table (4-12): Performance metric for XGBoost direct 1.....	83
Table (4-13): the hyperparameters for XGBoost direct 2.....	85
Table (4-14): Performance metric for XGBoost direct 2.....	85
Table (4-15): the hyperparameters for MLP direct 1.....	89
Table (4-16): Performance metric of MLP direct1.	90
Table (4-17): the hyperparameters for MLP direct 2.....	93
Table (4-18) Performance metric of MLP direct 2	93
Table (4-19): the hyperparameters for GRNN direct 1.....	98
Table (4-20): Performance metric of GRNN direct 1.....	98
Table (4-21): the hyperparameters for GRNN direct 2.....	102
Table (4-22): Performance metric for GRNN direct 2.	102
Table (4-23): the hyperparameters for Transformer direct 1.....	107
Table (4-24): Performance metric for Transformer direct 1.....	107
Table (4-25): the hyperparameters for Transformer direct 2.....	109
Table (4-26): Performance metric for Transformer direct 2.....	109

List of Figures

Figure (2-1): Chlorine Zones.....	12
Figure (2-3): Artificial Intelligence.....	16
Figure (2-4): Random Forest concept	18
Figure (2-5): Architecture of GRNN From	20
Figure (2-6): Architecture of MLP.....	21
Figure (2-7): XGBoost concept	22
Figure (3-1) Methodological Framework.....	31
Figure (3-2): Al-Hussein Neighborhood Water Project	32
Figure (3-3) GIS _Pipe line in the DWDN	33
Figure (3-4) Pipe line in the DWDN	34
Figure (3-5) Sample collecting bottles	35
Figure (3-6) Sketch of Pipe line 600mm in the DWDN	35
Figure (3-7) Sketch of Pipe line 500mm in the DWDN	35
Figure (3-8): Residual Chlorine tools and test.	37
Figure (3-9): PH test.....	38
Figure (3-10): TDS and EC test.	39
Figure (3-11): Turbidity meter.	40
Figure (3-12) Chloride test.....	41
Figure (3-13) Sulphate test.....	42
Figure (3-14): Total Hardness test.	43
Figure (3-15) Calcium test	44
Figure (3-16) Alkalinity test.....	45
Figure (3-17) Box plot.....	46
Figure (3-18): The Python icon and Anaconda logo.....	48
Figure (3-19): Transformer Model Architecture.	56

Figure (4-1): Distribution of PH and Standards (A) direct 1 (B) direct2.....	59
Figure (4-2): Distribution of Turbidity in direct1 for (A) P0(B)P1(C) P2(D) P3.	60
Figure (4-3):Distribution of Turbidity in direct 2 for (A)P0 (B)P1 (C)P2(D) P3.	62
Figure (4-4): Distribution of RC in direct 1 for (A) P0 (B) P1 (C) P2 (D) P3.....	64
Figure (4-5): Distribution of RC in direct 2 for (A) P0 (B) P1 (C) P2 (D) P3.....	65
Figure (4-6): Distribution of TDS and Standards (A) direct 1 (B) direct2.	66
Figure (4-7): Distribution of TH and Standards (A) direct 1 (B) direct2.....	67
Figure (4-8): Distribution of Ca and Standards (A) direct 1 (B) direct2.....	67
Figure (4-9): Distribution of Cl and Standards (A) direct 1 (B) direct2.	68
Figure (4-10): Distribution of So ₄ and Standards (A) direct 1 (B) direct2.....	68
Figure (4-11): Distribution of Temp and Standards (A) direct 1 (B) direct2.....	69
Figure (4-12): Distribution of EC and Standards (A) direct 1 (B) direct2.....	70
Figure (4-13): Distribution of Alk and Standards (A) direct 1 (B) direct2.	70
Figure (4-14): Comparison of Actual and Predicted Values for RC using RF S1 direct1.....	78
Figure (4-15): A Comparison of Actual and Forecasted Values for RC using RF S2 direct1.....	78
Figure (4-16): Comparison of Actual and Predicted Values for RC using RF S3 direct1.....	79
Figure (4-17) Comparison of Actual and Predicted Values for RC direct 2_ S1.	80
Figure (4-18) Comparison of Actual and Forecasted Values for RC direct 2_ S2	80
Figure (4-19) Comparison of Actual and Predicted Values for RC direct 2_ S3...	81
Figure (4-20): Comparison of Actual and Forecasted Values for RC using XGBoost S1direct1.....	84
Figure (4-21): Comparison of Actual and Forecasted Values for RC using XGBoost S2 direct1.....	85

Figure (4-22): Comparison of Actual and Forecasted Values for RC by XGBoost S3 direct1.....	85
Figure (4-23): Comparison of Actual and Forecasted Values for XGBoost direct 2_S1.....	86
Figure (4-24): Comparison of Actual and Forecasted Values for XGBoost direct 2_S2.....	87
Figure (4-25) Comparison of Actual and Forecasted Values for XGBoost direct 2_S3.....	87
Figure (4-26): Comparison of Actual and Forecasted Values for RC by MLP for S1 direct1.....	91
Figure (4-27): Neural Network Structure (MLP) for S1direct1.....	91
Figure (4-28): Comparison of Actual and Forecasted Values for RC by MLP for S2 direct1.....	92
Figure (4-29): Neural Network Structure (MLP) for S2 direct1.....	92
Figure (4-30): Comparison of Actual and Forecasted Values for RC by MLP for S3 direct1.....	93
Figure (4-31): Neural Network Structure (MLP) for S3 direct1.....	93
Figure (4-32): Comparison of Actual and Forecasted Values for RC by MLP for S1 direct2.....	94
Figure (4-33): Neural Network Structure (MLP) for S1 direct2.....	95
Figure (4-34): Comparison of Actual and Forecasted Values for RC by MLP for S2 direct2.....	95
Figure (4-35): Neural Network Structure (MLP) for S2 direct2.....	96
Figure (4-36): Comparison of Actual and Forecasted Values for RC by MLP for S3 direct2.....	96
Figure (4-37): Neural Network Structure (MLP) for S3 direct2.....	97
Figure (4-38): Actual vs Predicted Values for RC using GRNN for S1 direc1..	100

Figure (4-39): Generalized Regression Neural Network (GRNN) Structure for S1 direc1.....	100
Figure (4-40): Actual vs Predicted Values for RC using GRNN for S2 direc1..	101
Figure (4-41): Generalized Regression Neural Network (GRNN) Structure for S2 direc1.....	101
Figure (4-42): Actual vs Predicted Values for RC using GRNN for S3 direc1..	102
Figure (4-43): Generalized Regression Neural Network (GRNN) Structure for S3 direc1.....	102
Figure (4-44): Comparison of Actual and Predicted Values for GRNN S1 direct 2.....	103
Figure (4-45): GRNN structure Scenario 1 direct 2.....	104
Figure (4-46): Comparison of Actual and Forecasted Values for GRNN S2 direct 2.....	104
Figure (4-47): GRNN structure Scenario 2 direct 2.....	105
Figure (4-48): Comparison of Actual and Forecasted Values for GRNN S3 direct 2.....	105
Figure (4-49): GRNN structure Scenario 3 direct 2.....	106
Figure (4-50): Comparison of Actual and Forecasted Values by Transformer S1 direct 1.....	108
Figure (4-51): Comparison of Actual and Forecasted Values by Transformer S2 direct 1.....	109
Figure (4-52): Comparison of Actual and Forecasted Values by Transformer S3 direct 1.....	109
Figure (4-53): Comparison of Actual and Forecasted Values by Transformer S1 direct 2.....	110
Figure (4-54): Comparison of Actual and Forecasted Values by Transformer S2 direct 2.....	111

Figure (4-55): Comparison of Actual and Forecasted Values by Transformer S3 direct 2.....	111
Figure (4-56):The bar chart reflects the variation of the models in S1,direct 1..	115
Figure (4-57): The bar chart reflects the variation of the models in the S2, direct 1.	116
Figure (4-58):The bar chart reflects the variation of the models in the S3, direct 1.	116
Figure (4-59): Comparison of models of the S1 direct 2.	117
Figure (4-60): Comparison of models of the S2 direct 2.	117
Figure (4-61): Comparison of models of the S3 direct 2.	118

List of Abbreviations

AI	Artificial Intelligence
Alk	Alkalinity
ANN	Artificial Neural Network
BERT	Bidirectional Encoder Representations from Transformers
Ca	Calcium
Cl	Chloride
ClO₂	Chlorine dioxide
DBPs	Disinfection byproducts
DW	Drinking Water
DWDN	Drinking Water Distribution Network
EC	Electrical Conductivity
GPT	Generative Pre-trained Transformer
GRNN	General Regression Neural Network
HAAs	Haloacetic Acids
HOCl	Hypochlorous acid
IQR	Interquartile Range
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multi-Layer Perceptron
MLR	Multivariate Linear Regression
MSE	Mean Square Error
NCl₃	trichloramine

NH₂Cl	monochloramine
NHCl₂	dichloramine
OCI⁻	Hypochlorite ion
RC	Residual Chlorine
RF	Random Forest
RMSE	Root Means Square Error
SO₄	Sulfate
SVR	Support Vector Regression
TDS	Total Dissolved Solids
Temp	Temperature
TH	Total Hardness
THM	Trihalomethane
TOC	Total Organic Carbon
Turb	Turbidity
UV	Ultraviolet
WHO	World Health Organization
XGBoost	Extreme Gradient Boosting

List of Symbols

μ	Mean
σ	Standard deviation
K_b	Bulk decay constant
K_w	Wall decay constant

Chapter One: INTRODUCTION

1.1 Overview

Drinking water (DW) is considered an important human requirement, where contamination of water sources, due to geological reasons as well as agricultural and industrial uses, causes health hazards (**Sakran et al., 2019**). In 2006, the World Health Organization (WHO) reported that about 250,000 persons worldwide died daily from diseases such as cholera, attributable to the unsafe utilization of water. During recent years, the quality of water used for drinking has notably deteriorated as a result of pollutants. Monitoring the purity of water is important for the safety of humans, as contaminants can pose serious threats to health (**Ainapure et al., 2023**). Purifying DW involves multiple stages including screening large particulates, introducing coagulants, flocculating particles, sedimenting them, filtering, and disinfecting within the water treatment plant. The disinfection process represents an essential aspect of DW treatment. It is capable of avoiding waterborne illnesses and to inhibiting the spread of pathogens (microorganisms) within the water transport pipelines, from treatment facilities to consumers taps; nevertheless, it may also present significant health hazards, involving birth defects, genetically toxic effects, and carcinogenesis (**Sun et al., 2019; Jeffrey Yang, Haught and Goodrich, 2009**). Various disinfection techniques like ozone, UV light, and chlorination are effective but fail to maintain water quality during transportation across the water distribution system. Chlorination is a chemical disinfection technique employed to ensure the maintenance of residual chlorine concentrations within drinking water distribution networks (DWDN), ensuring safe and hygienic distribution until the last consumer

(Soyupak et al., 2011). Chlorine a common disinfectant utilized in DW treatment, efficiently inactivating harmful bacteria and assuring the safe transport of water through distribution pipelines **(Jeffrey Yang, Haught and Goodrich, 2009)**. While chlorination is a beneficial technique, but excessive doses can cause stomach injury and health issues due to the generation of disinfection byproducts like trihalomethanes and chloroform **(Abdulwahid, 2020)**. Furthermore, consumer exposure to high chlorine levels heightens the cancer risk and contributes to odor issues **(Soyupak et al., 2011)**. Thus, it is crucial to balance the objectives of sustaining appropriate chlorine residual levels to reduce microbial presence while preventing excessive concentrations that affect drinking water aesthetics. So, the WHO decided that DW must be free of organisms or germs, with permissible residual chlorine levels varying from 0.3 to 0.5 mg/l. Supervisors of the water purification facilities frequently regulate chlorine dosage based on residual chlorine (RC) and raw water quality data within the Water Discharge System (WDS). This approach frequently causes time delays because of the water travel time from dosing and measuring locations. The problem has resulted in an increase in efforts to simulate the residual chlorine levels in DWDN, solving the challenge of delayed action **(Bowden et al., 2006)**. The capacity to provide high-quality water production depends upon the management of the water system **(Susanto et al., 2020)**. Because the traditional approach to measuring chlorine leads to delays that diminish disinfection effectiveness **(Soyupak et al., 2011)**. Despite the progress in modern technology, it remained difficult to achieve the optimal level due to variable demand, old infrastructure, and inconsistent water quality across different regions. Also, the variation in water quality can be attributed to a variety of sources, which is a significant factor to consider.

Therefore, it is very important to deal with advanced techniques to effectively manage and improve the water quality by simulations for the important parameters that are effective on water quality. That's why researchers have conducted several studies to find artificial intelligence algorithms that can develop accurate models efficient at detecting complex trends in water quality data to forecast free RC levels at critical locations within the DWDN.

These models include deep learning and machine learning algorithms for estimating RC levels (**Onyutha, 2022**). The city of Karbala continues to utilize conventional methods for monitoring residual free chlorine. Due to the city's position as a popular religious tourism location, water consumption is elevated, complicating the issue more seriously. This means it is essential to find modern methods for maintaining and dealing with water quality. What encouraged the initiation of this thesis was the creation of strong models that aim to monitor and predict the RC in the selected points in the network to improve the RC by developing AI algorithms to forecast the free RC levels in the DWDN of the Al-Hussein neighborhood water project in Karbala. and compare models to identify the model exhibiting optimal performance.

1.2 Research Goals

The city of Karbala continues to utilize conventional methods for monitoring residual free chlorine. Due to the city's position as a popular religious tourism location, water consumption is elevated, complicating the issue more seriously. This highlights the necessity of finding modern approaches to maintain and manage water quality effectively.

The purpose of this study is to enhance the monitoring and control of the water quality in the Al-Hussein neighborhood water plant in Karbala by developing and comparing AI models to identify the most suitable one for

predicting the residual free chlorine at selected points within the DWDN and the plant point. The developed model must be capable of supporting decision-making to maintain chlorine levels within the permissible limits set by the WHO (2017) and the Iraqi DW standards No. 417 (2009). This ensures that there is no delay in dosing and that no malfunctions occur that could lead to problems such as network corrosion and the production of secondary disinfection materials in case of excess remaining chlorine, as well as the risk of insufficient disinfection dosage in case of low remaining chlorine. Additionally, this study aims to investigate whether the existing water quality parameters comply with these standards.

1.3 Research Objectives

1. Collecting and analyzing water samples from critical points within the distribution network to build a database for training the models.
2. Comparing the measured water quality parameters with the WHO (2017) and the Iraqi DW standards No. 417 (2009).
3. Using the augmenting technique to maximize dataset.
4. Creating AI models including Machine Learning (ML) and artificial neural networks (ANN).
5. Training AI models
6. Comparing the performance of the models based on standard criteria.
7. Extracting the best model for modelling and predicting residual chlorine concentration.

1.4 Thesis structure

Chapter One: presents general background of the study, the statement of the research problem, and the research goals.

Chapter Two: reviews the literaturer related to the topic of the thesis, as well as the definition of key terms.

Chapter Three: represents the methodology followed in this thesis in terms of the practical aspect and the programming aspect.

Chapter Four: discuss the AI Model results and discussion

Chapter Five: concludes a summary of key findings as well as the recommendations for future works.

Chapter Two: LITERATURE REVIEW

2.1 Introduction

The purpose of this chapter is to focus on the most significant prior research associated with the thesis topic, including an overview of DW quality, the most important disinfectants used for DW, the function of chlorine during the disinfection stage, as well as previous research associated with the employing of hydraulic modeling techniques for forecasting RC and water quality, and the most significant studies based on AI algorithms to predict DW quality, as well as the theoretical aspect of the algorithms used in this thesis.

2.2 General

The issue of water potability is difficult due to numerous contamination sources, primarily resulting from human activities **(Chinnappan *et al.*, 2023)**. According to the guidelines established by WHO, potable water must be free from any pathogenic germs or other bacterial forms **(Fardami *et al.*, 2019)**. When there's negative waste management, discharge from industries, and excessive utilization of limited water resources, the viruses, germs, and heavy metals, along with nitrates, enter water systems and make them polluted **(Jebril *et al.*, 2020)**. The preservation of water quality is therefore achieved through the application of safety protocols intended to sustain and enhance the purity of the water supply **(Organization, 2004)**.

2.3 Water treatment plants

The primary purpose of the water treatment is to provide consumers with safe DW. Treatment involves inhibiting bacteria, eliminating organic material that is naturally existing, avoiding harmful compounds, and maintaining aesthetic quality, as well as saving the distribution network from recontamination and corrosion. Physical-chemical and chemical processes

like coagulation-flocculation, settlement, particle filtration, and disinfection using chemicals form a substantial portion of traditional water treatment systems (**Bonton *et al.*, 2012**).

2.3.1 Coagulation

Coagulation includes introducing a chemical coagulation agent or coagulant to prepare colloidal particles, dissolve particulates, and suspend then for further treatment. This process often happens within a time frame of less than 10 seconds. Introducing a coagulant into water results in the instability and subsequent aggregation of the particles, known as flocculation (**Crittenden *et al.*, 2023**).

2.3.2 Flocculation

The process of flocculation typically occurs within a time frame of 20 to 45 minutes. During this stage, particles that have undergone destabilization and precipitation because of constituted from the addition of coagulants that aggregate to create bigger particles known as flocculent substances, or simply "floc.". The combined mass of particles may then be eliminated by the process of gravitational sedimentation as well as filtering (**Crittenden *et al.*, 2023**).

2.3.3 Sedimentation

The process of sedimentation is employed to minimize turbidity and particles that are suspended. Sedimentation tanks are engineered to lower the flow velocity in the water, allowing suspended solids to drop by gravitational force. Sedimentation tanks, often rectangular with length-to-width ratios varying from 2:1 to 5:1, are employed to eliminate substantial or dense particles, necessitating storage tanks to equilibrate demand (**DEFRA, 2018**).

2.3.4 Filtration

Diverse filtering methods are utilized in the purification of water for consumption. The filtration process serves as a reliable and efficient barrier against microbiological infections (**LeChevallier and Au, 2004**). To eliminate organisms present in water. It is commonly utilized to purify water for consumption in impoverished nations. The process involves the passage of water through permeable structures or membranes. The removal of suspended matter is achieved by processes such as compression, adsorption, sedimentation, and coagulation of the water. Filtration by membranes effectively eliminates pathogens by preventing the passage of germs that exceed the size of the membrane pores (**Pichel et al., 2019**).

2.3.5 Disinfection

Disinfection is the final stage of DW purification, involving the use of disinfectants to eliminate organisms that may cause disease. Its purpose is to guarantee that the water is safe to consume (**Gelete et al., 2020**). Chemicals, radiation, and heat are the three main cleaning technologies used in the treatment of water. Many disinfection chemicals are typically employed in DW treatment: free chlorine, combined chlorine, and chlorine dioxide, as well as ozone and ultraviolet radiation (UV). As for the UV light, it uses electromagnetic radiation; the first four are chemical oxidants (**Viera Valencia and Garcia Giraldo, 2019**).

2.3.5.1 Disinfection with chlorine

The process of chlorination was introduced at the start of the 20th century as an extra measure to protect against disease-causing germs (**Viera Valencia & Garcia Giraldo, 2019**). Chlorine is both an essential chemical for disinfecting water during treatment and a residual disinfectant to maintain the

quality of water throughout distribution. Despite that, by-products of disinfection are generated by the interaction of decontamination chemicals with organic or inorganic molecules, and exposure to these by-products might have detrimental effects on individuals' health (**EPA, 2013**). There are two types of chlorine used for disinfection:

a) Chlorination (Free Chlorine)

Chlorination is one of the most common methods for disinfecting DW. It involves introducing chlorine or chlorine compounds, such as calcium or sodium hypochlorite, into the water. The germ-inhibiting compound produced, hypochlorous acid, is often broadly referred to as “chlorine.” Once added, chlorine reacts in water to generate hypochlorous acid (HOCl) along with the hypochlorite ion (OCl⁻), which together are defined as free chlorine. This free chlorine acts as a powerful disinfectant through oxidation, particularly effective in clear water with a pH below 8.0 (**EPA, 2013; Pichel et al., 2019**). The most significant benefit of chlorination is that when added to water at the proper dosage, chlorine stays as residual chlorine, which maintains the ability to disinfect water throughout the distribution and storage networks (**WHO, 2017**). However, while it's successful in disinfecting water, chlorination has disadvantages including a disagreeable taste and odor, as well as its ineffectiveness in thoroughly eradicating protozoan cysts and eggs (**WHO, 2017; EPA, 2013**). Furthermore, when disinfecting, chlorine can combine with organic materials to create other disinfection byproducts, like trihalomethanes (THMs) and haloacetic acids (HAAs) (**Hao et al., 2017**). The optimal and secure RC amount in a DW system ranges between 0.2 and 0.5 mg/L, as defined by the WHO (**Organization, 2022**).

b) Chloramination (Combined Chlorine)

Chloramination treatment involves the deliberate introduction of ammonia and chlorine to cause a reaction and form monochloramine (NH_2Cl). This reaction must take place at the purification place (**Pichel et al., 2019**). Its disinfection efficacy is weaker than chlorination, hence it is often used as a supplementary disinfectant during transportation and not as the primary disinfectant (**EPA, 2013**). Chloramination necessitates the use of comparable dosing equipment and professional employees as chlorination; however, it generates fewer taste and odor problems and does not result in the formation of THMs (**Pichel et al., 2019**). One of the best things about monochloramine is that it doesn't produce THMs or other by-products of the chlorination Process when interacting with organic matter. Its taste threshold is often higher than chlorine alone, making it a more suitable alternative for reducing customer discomfort with chlorine taste (**EPA, 2013**).

2.3.5.1.1 Chlorine dioxide

Chlorine dioxide (ClO_2) has greater oxidizing capabilities compared to chlorine and chloramines. The disinfection behavior of this chemical is less affected by pH levels than chlorine and it has a durable residual activity. Due to its instability, chlorine dioxide must be created at the location by combining chlorine or acid with sodium chloride, and then adding the mixture to the water. This procedure needs continuous care, monitoring, and supervision, and it is significantly more costly than chlorination (**WHO, 2017**). The disinfection byproducts (DBPs) that are produced during water purification with Chlorine dioxide were shown to produce chlorite and chlorate ions, which are inorganic compounds. Consequently, chlorine dioxide was mostly used in treating low Total Organic Carbon (TOC) waters

that required only a small quantity to satisfy their oxidant requirements **(Crittenden *et al.*, 2023)**. Both chlorites and chlorates lead to the occurrence of methemoglobinemia even hemolytic anemia in consumers **(Al-Otoum *et al.*, 2016)**. Nevertheless, the levels of THMs and HAAs in water are lower if ClO_2 is used instead of chlorine. However additional drawbacks of ClO_2 include its limited efficacy against *Cryptosporidium* oocysts and its propensity to cause taste and odor problems. These factors restrict its use as a supplementary disinfectant **(Pichel *et al.*, 2019)**.

Chlorine dioxide has more efficacy as a disinfectant compared to chlorine when the pH level increases. Chlorine dioxide is expected to have a much higher cost compared to chlorine **(EPA, 2013)**.

2.3.5.1.2 Chlorine chemistry

When chlorine gas is added to water, it quickly interacts and then dissolves to produce hypochlorous and hydrochloric acid. As in the equations below



A strong acid that dissolves and lowers pH and alkalinity is hydrochloric acid. Hypochlorous is considered a weak acid, and pH influences its dissociation. In water, chlorine and ammonia reactions generate chloramines. As chlorine increases, three chloramine species arise. As shown in equation (2-3), the first type is monochloramine; equation (2-4) shows dichloramine; and equation (2-5) shows trichloramine.





The culmination of these three results of reaction is termed combined chlorine. Total chlorine is equal to the overall amount of free chlorine, together with combined chlorine (**Crittenden *et al.*, 2023**). Hypochlorous acid is a very powerful chemical for reducing bacteria in water (**Viera Valencia & Garcia Giraldo, 2019**).

2.3.5.1.3 Chlorine Zones

The three zones in Figure (2-1) illustrate reactions and their processes.

In zone A, in water, the overall RC quantity is increased to be approximately the same amount as the chlorine that is supplied, until the proportion of chlorine on ammonia molecules is close to 1, provided that there are no other sorts that deplete chlorine.

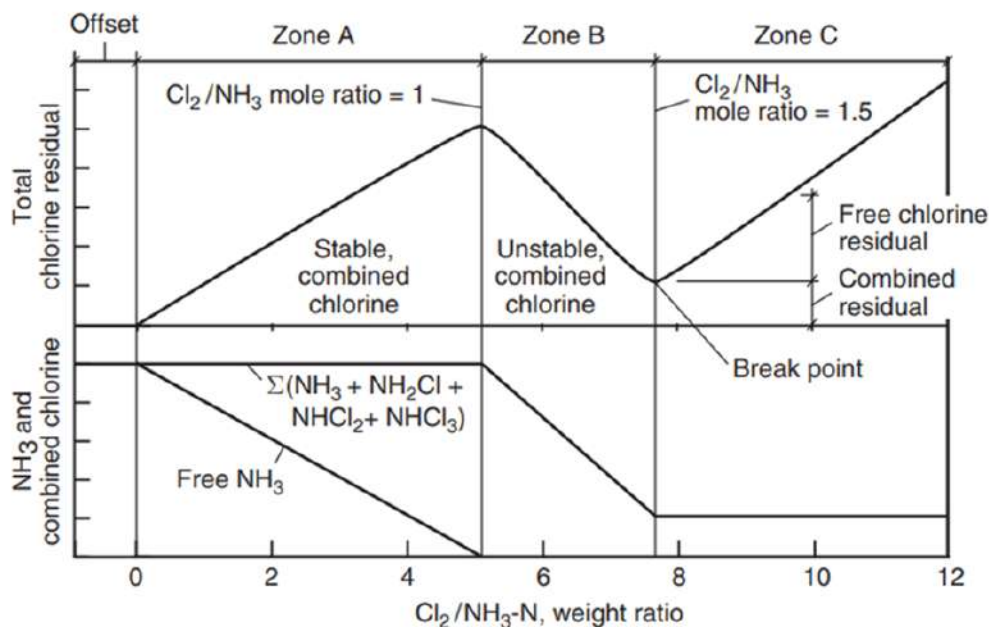
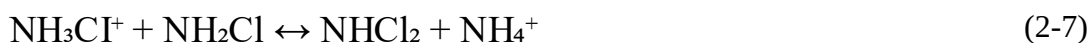


Figure (2-1): Chlorine Zones (Crittenden *et al.*, 2023)

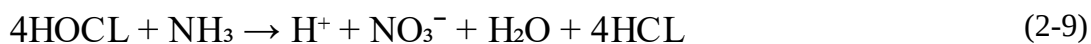
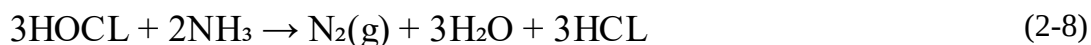
Monochloramine rapidly forms in zone A; however, the species in this zone is influenced by pH levels, where the pH decreases, resulting in the

formation of dichloramine through chemical interactions as shown in equations (2-6) and (2-7) (**Crittenden et al., 2023**).



At a pH of 8, monochloramine is the only type of chloramine found in zone A. However, significant levels of dichloramine may exist at a pH of 6 (**Palin, 1975**).

Zone B: Increasing the amount of chlorine leads to a reduction in the overall chlorine residue. When the molar ratio exceeds 1, it indicates that chlorine is causing the oxidation of certain chloramine species. In the end, almost all of the chloramine undergoes oxidation (**Crittenden et al., 2023**). Where dichloramine is present in zone B, characterized by higher chlorine levels, even when the pH is 8 (**Palin, 1975**). In this region, HOCl acted to oxidize ammonia, resulting in the creation of nitrogen gas and nitrate ions, meaning no chlorine remains as, illustrated in equations (2-8) and (2-9). This reaction results in the formation of nitrogen gas (**Saunier and Selleck, 1979**).



Zone C: starts at the breakpoint, which indicates the end of the chloramine species oxidation process. When the breakpoint is achieved, the amount of added chlorine provided directly correlates with an increase in the level of free chlorine (**Crittenden et al., 2023**).

2.3.5.2 Disinfection by ozone

Ozone considered the strongest chemical used for disinfectant. It is an unstable gas generated by the breakdown of oxygen molecules into individual

oxygen atoms, and has an unpleasant smell that is detectable by a lot of people in the atmosphere at concentrations higher than 0.1 ppm (**Crittenden et al., 2023**). Ozone is regarded as highly effective in eliminating viruses and is often considered more effective than chlorine. Where ozonation does not produce dissolved solids, and its efficacy is unaffected by ammonium ions or variations in pH. Consequently, ozonation is regarded as another option to chlorination (**Eddy et al., 2002**). It mainly used as a disinfectant to inhibit the growth of harmful bacteria and to remove iron, manganese, odor-causing chemicals, color, and resistant organic compounds. This type of disinfection is more expensive than chlorination or UV light disinfection. (**Viera Valencia and Garcia Giraldo, 2019**).

Ozone can be used in conjunction with post-chlorination for water treatment because of its non-residual characteristics in water (**Viera Valencia and Garcia Giraldo, 2019**). Gaseous ozone is produced at the location of the treatment facility before being injected into the water (**Crittenden et al., 2023**).

2.3.5.3 Disinfection by UV radiation

Ultraviolet radiation, a form of electromagnetic radiation, is employed for sterilization to inactivate germs in water, air, and some medical surfaces. It is generally acknowledged that bacteria are inactivated by ultraviolet (UV) radiation through changes to their DNA. However, the UV lights don't add chemicals and don't leave any effects behind when the water exits the UV reactor (**Bolton and Cotton, 2011; EPA, 2013**). The presence of chemical contamination and the creation of biological films contribute to the complexity and high cost of maintenance (**Brander, 2003**). Also, the presence of compounds in water may contaminate the outside surfaces of the lamp of

UV arms, leading to a decrease in the amount of ultraviolet light and disinfection impact (**EPA, 2013**).

2.3.6 Comparison of disinfecting techniques

Among the above disinfection methods, all processes demonstrate efficacy in eliminating pathogens. Due to ozone not being stable and UV depending on radiation directed in real time, these two methods are inadequate to disinfect water after it leaves the treatment plant. As for chlorine, although a chemical substance that induces secondary reactions, it is notable for its sustained disinfection efficacy, as it maintains a residual concentration that inhibits the regrowth of contaminants during water transport, thereby ensuring continuous safety across the DW distribution system until reaching consumers' taps. Furthermore, it is more economical than ozone and UV and is simple to apply (**Abbas, 2017**).

2.4 Artificial Intelligence (AI) approach

AI represents the ability of a computer to mimic human brain function. A subfield of AI is called machine learning (ML)(**Gupta, 2018**), it represents a domain of computer science that employs statistical methodologies and algorithms to enable an artificial system to learn by processing large quantities of data (**Swamynathan, 2019**). where algorithms are generally classified into two categories: unsupervised and supervised learning. The initial category contains techniques applicable to classification and regression tasks, dependent on the nature of both the input and the output variables (**EL Bilali et al., 2021**). ML enables machines to examine data and decision-making and improve results according to experience, and learn from provided information without direct programming (Chelliah, Sakthivel and Nagarajan, 2022). Deep learning is a branch of ML that uses algorithms

known as Artificial neural networks (ANNs). These networks simulate the behavior of biological neurons in the nervous system to process and transmit information (Chelliah, Sakthivel and Nagarajan, 2022), and they are capable of self-learn, being modeled after the anatomy and physiology of the brain. Instead of being specifically instructed on strategies to solve a problem, ANNs get trained to learn how to recognize concepts as well as patterns (**Gupta, 2018**), Figure (2-3) represent Artificial Intelligence (**Saikureti, 2020**).

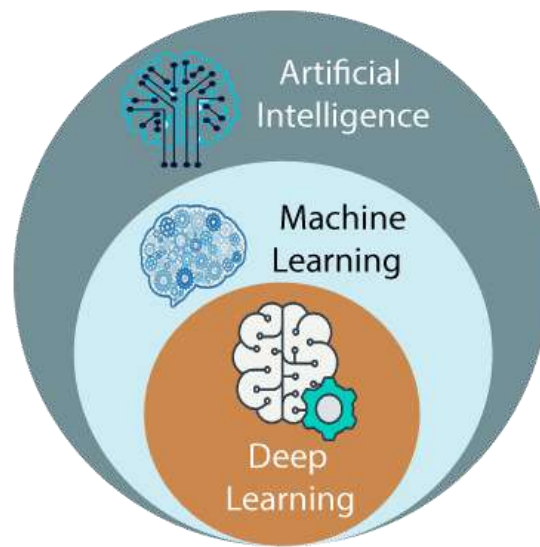


Figure (2-3): Artificial Intelligence (Saikureti, 2020).

ANN are a prevalent approach in AI. This approach is beneficial for developing predictive models that can manage complex connections among input parameters. It consists of interconnected nodes formed from user input, hidden layers, and output. The processing element makes a mathematical multiplication operation on its input data with many different weights and then transforms the result toward an output value in two ways: linearly or non-linearly to provide an outcome that corresponds with the measured output (**Wadkar et al., 2021**). Artificial networks, deep learning approaches,

decision trees, k-nearest neighbors, genetic algorithms, hybrid algorithms, and various other methodologies illustrate AI technologies. The implementation of these new technologies diminishes human labor. AI modeling techniques are essential in the sanitation and water purification processes, as automation has rendered these procedures easy, economical, and devoid of errors. Utilizing AI and deep learning for modeling gives an environmentally friendly and economical method for foreseeing and reporting faults and the exceptional cases arising from significant situations (**Aldahdouh, 2017**).

In this study, used five AI algorithms as follows:

2.4.1 Random Forest (RF)

RF is a group decision tree approach that utilizes the results from many equivalent weak decision trees to generate forecasts. The weak tree arbitrarily partitions a minor segment of the training data and input values, concluding in an overall RF that represents the average of the outputs of weak trees to achieve higher quality results (**Onyutha, 2022; Kyritsakas et al., 2023**). RF is a merger of multiple tree-structured classifiers. It has been extensively utilized in classification and prediction, as well as in regression. Furthermore, it has possessed numerous advantageous qualities in comparison to classic algorithms. As a result, RF includes a wide variety of applications. The method of forecasting is formed by averaging the final results from the different trees. Low-error trees could produce highly accurate regression forests. As well as being capable of using both categorical and numerical data (**Liu, et al., 2012**). These algorithms function based on the straightforward yet efficient "divide and conquer" concept: they sample sections of the data set, develop a randomly generated tree predictor for each subset, and subsequently aggregate all predictors. RF is mainly attributed to their adaptability in

forecasting diverse issues and their limited adjustment parameters. This method is recognized for its precision, capacity to manage small sample numbers, and highly dimensional feature spaces, with its straightforward parallelization rendering it appropriate for extensive real-world systems **(Biau & Scornet, 2016)**.

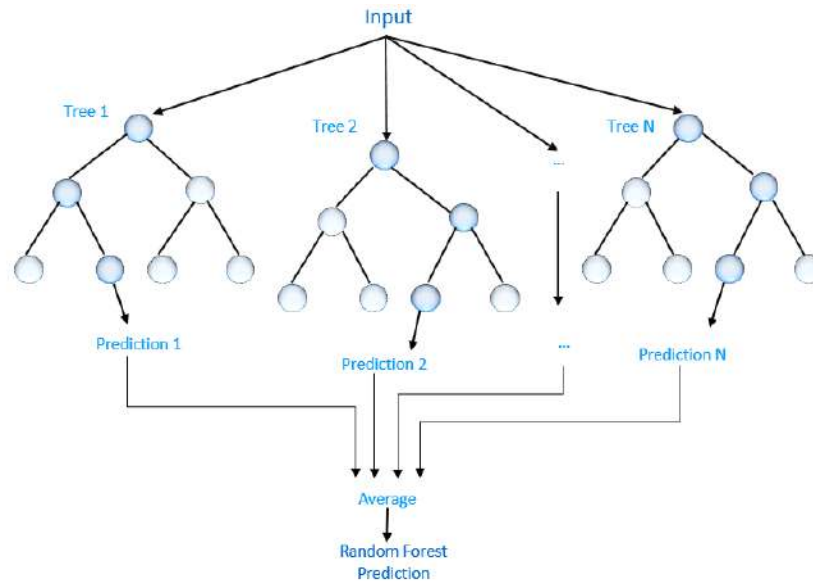


Figure (2-4): Random Forest concept (Sharma, 2022).

2.4.2 General Regression Neural Network (GRNN)

The scientist Specht created the artificial neuron known as GRNN for feedforward networks **(Specht, 1991)**. It represents a particular type of neural network. It has demonstrated efficacy for addressing a number of scientific and engineering problems **(Leung et al., 2000)**. The operation of GRNN fundamentally relies on the concept of nonlinear regression for calculating the predicted output value based on the provided input data. GRNN can generate multivariate output vectors. This network architecture is characterized by rapid learning and the utilization of a simple and direct training process. Additionally, it can handle errors, including outliers, which makes it useful

for dealing with inaccurate information. The GRNN may approximate any random function based on previous data (**Leung et al., 2000**).

The following equation illustrates the GRNN concept in a corresponding nonlinear regression equation:

$$\hat{Y}(X) = \frac{\left[\sum_{i=1}^n Y^i \exp \left(-\frac{D_i^2}{2\sigma^2} \right) \right]}{\left[\sum_{i=1}^n \exp \left(-\frac{D_i^2}{2\sigma^2} \right) \right]} \quad (2-10)$$

Where $\hat{Y}$ equal to predicted value

Y^i represent the actual data for training sample

D^2 represents the Euclidean distance through the new input and the data points within the training set.

$$D^2 = (X - X^i)^T (X - X^i) \quad (2-11)$$

σ is equal to the smoothing parameter (**Bowden et al., 2006**).

The GRNN does not require a recurrent training process, in contrast to the backpropagation approach. Moreover, it is evident that as the training set's quantity increases, the calculated error converges to zero. It relates to a conventional statistical method known as kernel regression (**Cigizoglu, 2005**). The architecture of GRNN consists of an input layer that intended to receive the input values. In the network, the number of cells is similar to the number of input variables. The following layer, called Pattern Layer, uses a cell to represent each training sample. The distance between each training sample and the new inputs is defined by the Euclidean distance, which is computed in this layer. The Gaussian function is then given the estimated distance. The summation layer is the third layer, which is made up of nodes of both types. Use the A-Summation Node to add up the (weighted) output values. The B-

Summation Node is used to add up the weights. The last layer represents the output layer, which displays the model's outputs as determined by the GRNN equation. Figure (2-5) shows the architecture of GRNN (**Bowden et al., 2006**).

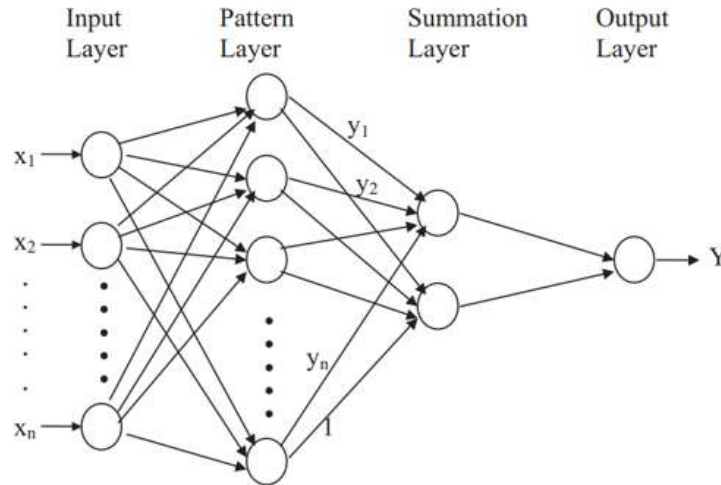


Figure (2-5): Architecture of GRNN From (**Bowden et al., 2006**).

Determining the ideal value for the smoothness factor σ , which is usually achieved by trial and error, is necessary to achieve high GRNN efficiency because σ is a critical and important factor (**Bowden et al., 2006**).

2.4.3 Multi-Layer Perceptron (MLP)

It is a subset of deep learning that represents part of machine learning, which depends on artificial neural networks (**Lashgari et al., 2020**). while learning under supervision, containing multiple neuronal layers. Usually, each neuron in the following layer is coupled with its output. Hidden layers are positioned between the input layer and the output layers, MLPs may learn complex patterns that cannot be split by linear and other easily expressed curves, making them useful for pattern classification, recognition, forecasting, and approximation. As more neurons more layers develop, an MLP network's

ability to learn complex patterns enhances. MLPs have demonstrated success in a number of AI applications. An algorithm is trained by optimizing efficiency on a particular dataset and memorizing training samples. Before the learning process begins, the value of a hyperparameter (like the amount of layers, and neuron for every layer, the type of activation function, the initial quantity of weights, the rate of learning, etc.) is determined. The effectiveness of the network can be significantly influenced by slight variations in the values of specific parameters (**Gupta, 2018**). MLP was employed to deal with many ML challenges throughout many domains since to its efficacy in forecasting each continuous and categorical variables (**Fesaghandis et al., 2017**).

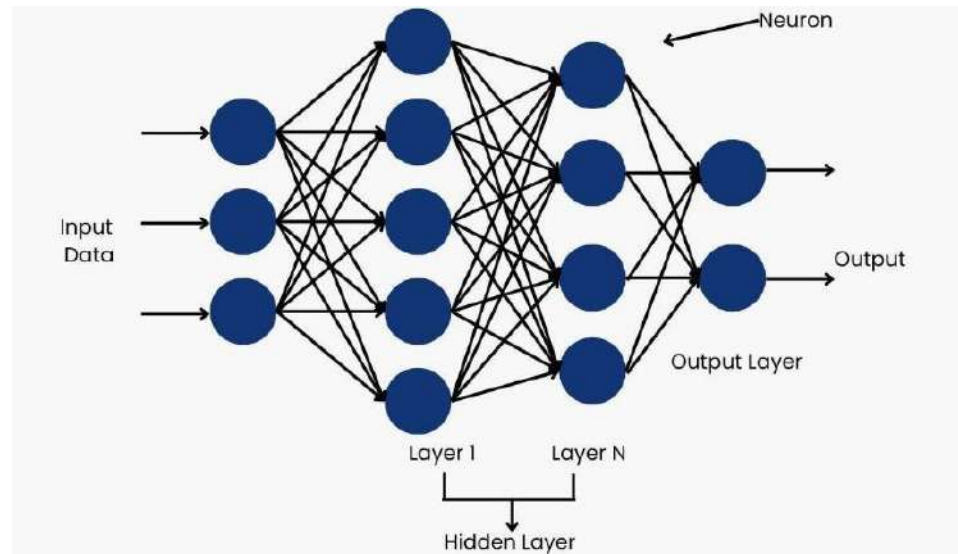


Figure (2-6): Architecture of MLP (**Turing, 2022**).

2.3.6 Extreme Gradient Boosting (XGBoost) algorithm

XGBoost is the most powerful and modern ML algorithm that demonstrates exceptional capabilities in building complicated systems (**Ali et al., 2023**). It also consists an aggregation of several decision trees, where each newly built tree fixes the errors of its predecessor, thus improving the model's efficacy. This approach possesses the ability to deal with missing information

throughout training. Also, the introduction of a regularization term controls the level of complexity in the model. subsequently improves generalization abilities with different data and reduces overfitting. Moreover, it can handle data with large dimensions (**Zhou et al., 2019**).

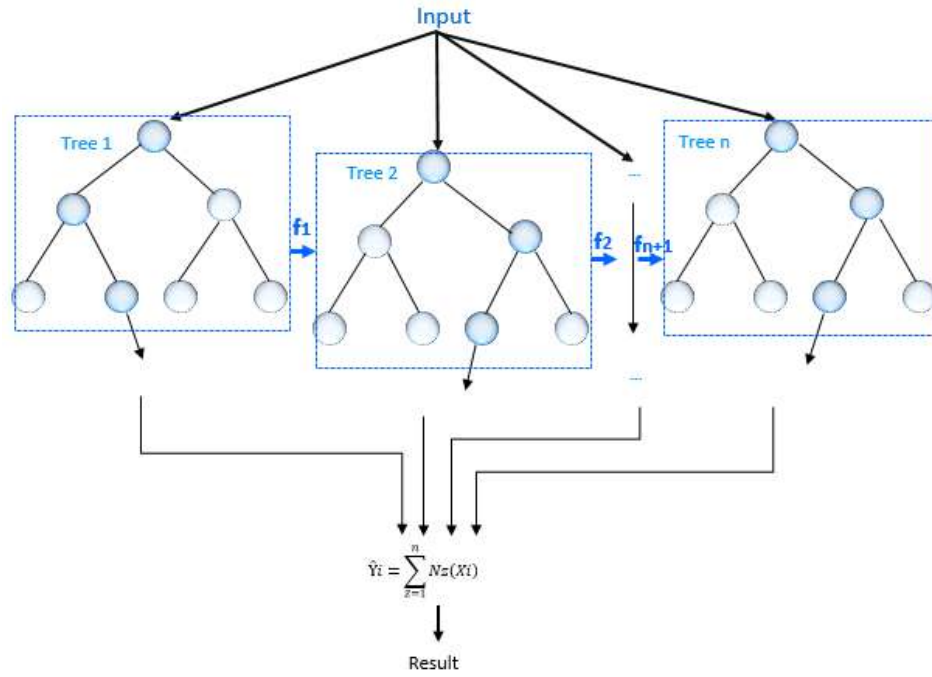


Figure (2-7): XGBoost concept (**Wang et al., 2019**).

$$\hat{Y}_i = \sum_{z=1}^n N_z(X_i) \quad (2-12)$$

$\hat{Y}_i$ represents the predicted probable value of the test sample X_i .

N_z represents the tree number z (**Wang et al., 2019**).

2.4.5 Transformer Model

Transformers, a category of deep learning ANN, were initially presented by researchers from Google in the 2017 paper titled "Attention is All You Need." (**Vaswani et al., 2017**). This latest architecture rapidly brought about a revolution within the field of language processing. Such as

the Generative Pre-trained Transformer (GPT) and Bidirectional Encoder Representations from Transformers (BERT) models, which are based on the Transformer building structure and have significantly surpassed earlier neural networks (**Gillioz *et al.*, 2020**). This algorithm's advantage is found in its exclusive use of self-attention, allowing it to manage sequences with exceptional efficiency (**Cristina, 2023**). Since the fundamental architecture of the algorithm is based on two layers identified as the encoder, which converts the input data into an internal representation that the model recognizes, along with the Decoder layer that generates the outputs, and both include multiple accumulated layers whose outcomes are independent, referred to as Independent features (**Vaswani *et al.*, 2017**). This algorithm has demonstrated its effectiveness and strength in various recent research. As observed in the study by (**Galindo Padilha *et al.*, 2022**) , where it serves as a hybrid model for forecasting multi-variable renewable energy. Also, in the field of neuroscience, where it was applied to classify neuroimages, the model showed exceptional performance in each regression and classification applications (**Zhu *et al.*, 2025**). Due to its novelty, it is still under study in many fields because it is a promising algorithm from the researchers' perspective.

2.5 Previous Study

Bowden *et al.* (2006) used an ANN model in a case study in South Australia to predict remaining chlorine. The process gathers historical data on all variables that could affect chlorine decay as well as the remaining chlorine at primary WDS locations. This study aims to evaluate the long-term forecasting ability of the ANN model, identify significant variables, and contrast the GRNN against multiple linear regression (MLR) models.

Researchers found that the GRNN model effectively predicted RC in a DWDN and that a 24-hour forecasting horizon was suitable. According to the study's findings, GRNN models outperform MLR models in utilizing the nonlinear interactions present in the dataset.

Soyupak *et al.* (2011) The Konyaalti DWDN in Turkey were designated as an experiment. This project began to critically examine the efficacy as well as accessibility of the ANN technique as a method to regulate free RC concentration in complicated DWDN provided by high-quality sources with minimal chlorine requirements. The research demonstrates that ANN models with significant forecast accuracy can be created for DWDNs, and these models are suitable for predicting objectives. Adequate data collection is essential for the effective utilization of established ANN models as regulators for free RC levels in DWDN.

A study by **Wadkar & Kote (2017)** in India, due to the randomness in adding chlorine doses within the plant, necessitated the development of a model to estimate RC levels in the DW treatment plant. They developed thirty-two neural network models for four different input combinations; every model underwent training ten times. The data is divided into three data sets, where 70 percent has been allocated for training, 15 percent used for validation, and 15 percent for testing. The GRNN demonstrated markedly reduced errors for all data sets when compared with the MLR. It also shows that models with four inputs perform better in the networks than models with different input variations. GRNN models surpassed the other thirty-two models in predictive performance across multiple combinations of input and target variables. The GRNN 4 model had the lowest MSE of 0.001 and MAE of 0.019, as well as the greatest R^2 value of 0.979, when ANN model

performances were compared. Additionally, it was noticed that as the number of input variables increase, this improves the effectiveness of ANN models.

A research study performed by **Hameed *et al.* (2018)** in Kirkuk, Iraq, assessed the RC levels in DW at five locations. Samples were collected from each station throughout the initial seven months of 2017, utilizing several parameters including, temperature, turbidity, pH, total dissolved solids, and electrical conductivity for water testing. Research findings indicate a negative correlation exists between RC content and water temperature. The results demonstrated that all tests satisfied Iraqi regulations, with the exception of residual chlorine, which was found to be higher than permitted, with levels ranging from 0.8 to 3 mg/L.

Joslyn (2018) employed nine variables for water quality based on data taken every 30 minutes over an interval of three years. The author employed two ML methods, such as XGBoost and Support Vector Regression (SVR), for making predictions, built using Python. Each run selects different targets, representing the eight remaining variables as inputs. The data has been split in to 90% to be trained and 10% for testing. The SVR results indicated the maximum predictive performance for temperature as 98.43%, while the lowest accuracy identified was for turbidity at 81.45%. Each variable had an accuracy over 79%. In the same way, the XGBoost model possessed the highest predictive performance for temperature at 98.49%, while the lowest performance was recorded for pH at 81.64%. It achieved a performance over 80% across all factors. Overall, the SVR model surpassed XGBoost; nonetheless, XGBoost demonstrated superior speed of operation and yielded satisfactory results for the most parameters.

García-Ávila *et al.* (2021) focused on developing an EPANET model for the simulation of RC level in the DWDN, aiming to ascertain compliance with the WHO-recommended RC level as a strategy for minimizing COVID-19 in the Ecuadorian city of Azogues. For the improvement of disinfection using chlorine during pandemics. Thirty points for sampling were chosen to collect 30 monthly samples over six months. The temperature dependence of chlorine decay constants was discovered. The decay rate constant increases with temperature. For the research region, the mean magnitude of the constant k_b , which is 0.15 h^{-1} , was measured. The results suggest that the chlorine degradation rates along the wall are influenced by pipe diameters and flow velocities. Investigative shows indicate that the flow velocity and K_w have a positive correlation. Additionally, there is an important relationship between the velocity of flow in pipes and reaction rates involving the surface of PVC pipelines. The sensitivity test indicates that Bulk decay constant (K_b) is the primary factor contributing to residual chlorine decay, whereas Wall decay constant (K_w) has a lesser impact on it. A beginning chlorine level of 0.87 mg/L was found to be appropriate to sustain a residual level of approximately 0.3 mg/L within the system, according to the Ecuadorian requirement. In contrast, with a starting chlorine level of 0.64 mg/L , the result is less than 0.3 mg/L at 48% of the nodes. It may be concluded that the chlorine decay in this investigation was mainly caused by disinfectant interactions with the water mass.

In a previous investigation of **De Santi *et al.* (2021)** constructed ANN group models from data collected from 3 refugee camps. employed groups of ANNs to forecast the place of consumption free RC level and establish, the place of distribution of Free RC goals according to the hazard of insufficient

levels at the consumption location. The findings indicated that the algorithms perfectly predicted the risk of reduced place of consumption.

According to a study by **Mahato & Gupta (2022)**, the use of AI techniques and multivariate linear regression (MLR) in real-time forecast of trihalomethane (THM) compounds with greater accuracy in DW supplies in India over two seasons is necessary since there are elevated levels above WHO's required standards and because they are difficult to determine due to complicated procedures. MATLAB was utilized to apply the support vector machine, Python was used to create ANN, and SPSS software was used for MLR. The ANN model's findings were supportive, followed by those of the SVM and MLR models. Consequently, it is thought that the AI modeling methodology serves as an effective instrument for the real-time prediction of THM chemicals in water. In contrast to the conventional MLR method, this study demonstrated that AI algorithms can predict complex chemicals in drinking water, despite their concentration on THM rather than RC.

Belcaid et al. (2023) were selected the Moroccan city of Mohammedia as a study area. The majority of the 54% cast iron pipes demonstrated a possibly elevated blockage rate of 70%. The main emphasis of this research project implementation of an EPANET model in conjunction with a water purity model, and the monitoring of concentrations and injection rates over an adequate amount of time to estimate the chlorine level at a specific place inside the network. The decay model was calibrated using actual data about the residual amounts of chlorine in supply systems. The modeling findings show that iron-cast material pipes exhibit a substantial decrease in free chlorine. Chlorine is required in higher quantities for ductile iron pipes than for plastic materials. Therefore, when chlorine level is below 0.2 mg/L, it is

possible to identify consumer vulnerability zones and make adjustments to new drinking water network plans and procedures that aim to reduce the network's size. The most evident of these appears to be rechlorination in regions with extremely low quantities of chlorine.

2.6 Summary

Disinfection is a crucial stage in water treatment, with chlorine being the most common and cost-effective disinfectant, and its ability to leave residues, makes it dangerous if not maintained. according to the WHO (2017). Consequently, it is important to monitor RC levels, as previous studies have demonstrated many methodologies for assessing chlorine concentrations, including conventional hydraulic models and mathematical techniques. Recently, AI domains, such as ANNs and various ML algorithms, have been employed to model systems can predict chlorine levels. These models have demonstrated notable superiority over traditional models in water quality monitoring studies, particularly the GRNN algorithm, has proven effective in forecasting disinfectant levels. Considering the variation in water quality across different regions and the limited number of previous studies in this domain, there is currently no research employing AI algorithms to predict residual free chlorine levels in the DW network of Karbala. Furthermore, the previous study demonstrated that XGBoost is an excellent algorithm for forecasting water quality variables. However, studies on forecasting RC with XGBoost are nearly absent. Additionally, the Transformer algorithm is novel, and the researchers view it as a promising approach. And since the absence of its application for predicting RC in DW, this motivated the start of research to study the training of AI algorithms at specific points within the DWDN as well as the experiment of employing XGBoost to predict the RC and applying

the Transformer to contribute to advancing the science in this field, in other words, aiming to develop models that can forecast disinfectant levels at strategic locations. The purpose of this thesis was to fill these research gaps.

Chapter Three: Methodology

3.1 Introduction

This chapter focuses on clarifying the study approach employed in the current study. Where the document includes two sections, The first one explores the chosen study region, defining the pipes and sampling points, including their quantity and location within the DWDN, with supporting graphics. It involves the laboratory and field tests carried out, the devices utilized in the experiments, and the materials involved. Also, the second part of the chapter focused on the programming aspect, including the programming language utilized and the work environment in which the code was carried out. Furthermore, it involved the demonstration of selected AI models, including ML models and ANN, with the purpose of training them on actual data. Also explains the way to evaluate model performance. Additionally, techniques for handling, augmenting, and dividing data. This chapter looks into the experimental and technological methodologies employed to achieve results based on clear scientific principles that match the thesis goals. Figure (3-1) represent the methodological framework and the images are for illustrative purposes from the sources **(Rawpixel, 2023; iStock, 2024; Pngtree, 2024)**.

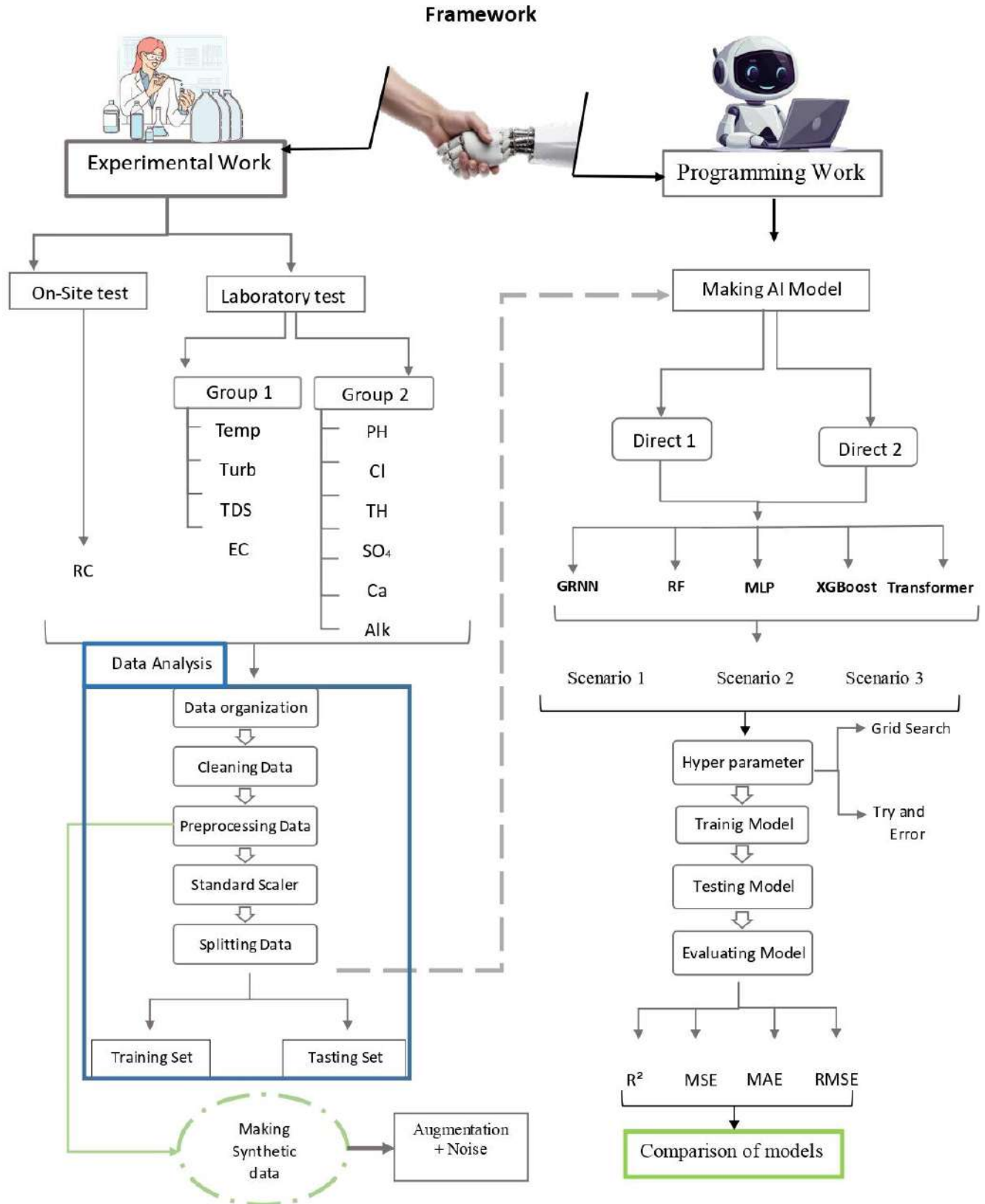


Figure (3-1) Methodological Framework

3.2 Research region

The research location was selected within the Al-Hussein neighborhood water plant. It is located in the city of Karbala, Iraq. According to information from the Karbala Water Directorate (2024), the plant was split in to two segments. The initial central water project started operating in 1973, with a capacity of about 2000 m³/hr, and the second project, comprising 10 combined water units, each with a capacity of about 200 m³/hr, also totaling 2000 m³/hr, began operations around 2005. Consequently, this study adopted the first one. Figure 3-2 illustrates that the water plant of Al-Hussein neighborhood is located in the center of the residential region. The plant is characterized by an excessive demand for water. It serves the nearby residential and commercial areas, which have a high population density, with water (**Ibraheem, 2024**).



Figure (3-2): Al-Hussein Neighborhood Water Project (**Al-Hussein neighborhood water project, 2023**).

3.3 Experimental Work

3.3.1 Method of data collection

DW samples were collected from June 23, 2024, to October 1, 2024, during the summer, the sampling was taken twice a week, with eight samples each time, for a weekly total of sixteen samples. The sample collection occurs from two water lines, the first one with a diameter of 600 mm (purple pipe) and the second with 500 mm (blue pipe), as illustrated by the GIS graphic from the Karbala Water Directorate. Figure (3-3) and Figure (3-4) show the sample points.

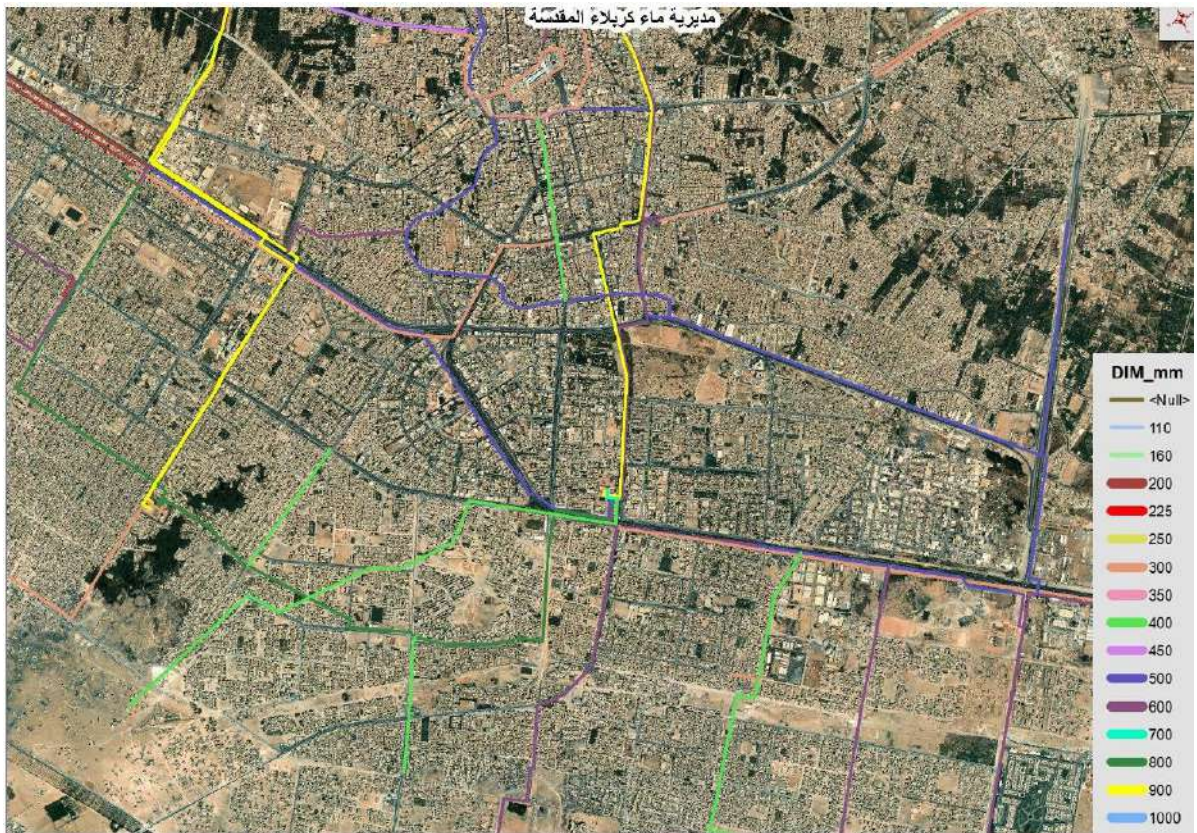


Figure (3-3) GIS _Pipe line in the DWDN from Karbala Water Directorate (2024).

Samples are collected in clean and locked plastic containers. Figure (3-5) reflects gathered samples. The total number of records is 116 samples each, equal to 232 samples overall. The project point and three points are situated

approximately at the beginning, midpoint, and end of the network. Figure (3-6) and Figure (3-7) exhibit the sampling points sketch on pipes one and two, respectively. The first direction's points, P0, P1, P2, and P3, are located at distances of 330, 850.5, and 1528 meters, respectively, whereas the second direction's points are 870, 1151, and 2196 meters distant. The distances were roughly computed utilizing the distance measurement tool on Google Earth.



Figure (3-4): Pipe line in the DWDN

A velocity assumed 1 m/s according to **(Richardson, 1989)**, the velocity of water in the pipe ranges (0.6 to 1.5) meters per second, as a reduced speed would result in sediment accumulation within the pipes, whilst an increased speed would induce pipe erosion. Therefore, the velocity is approximately assumed to be in between. After determining the distance and assuming the speed, then calculate the travel time was calculated using equation (3-1) **(Halliday et al., 2014)**.

$$\text{Velocity}(\text{m/sec}) = \frac{D}{T} \quad (3-1)$$

D indicates the distance in meters, whereas T represents the time in seconds.



Figure (3-5): Sample collecting bottles.



Figure (3-6): Sketch of Pipe line 600mm in the DWDN.

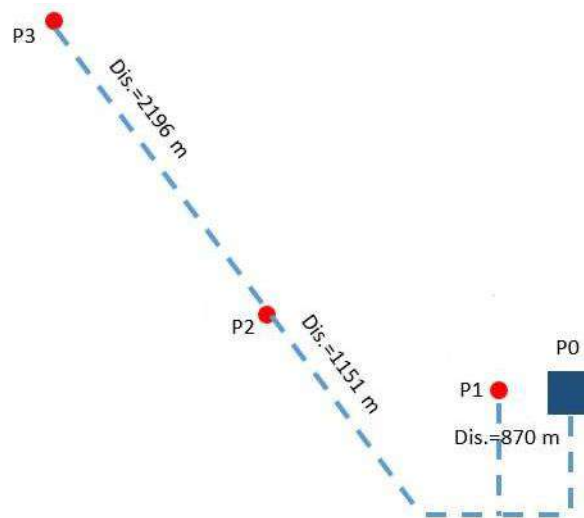


Figure (3-7): Sketch of Pipe line 500mm in the DWDN.

3.3.2 The measured variables

A series of physical tests, like turbidity, temperature, electrical conductivity, and total dissolved solids, and chemical tests, like alkalinity, pH, calcium, sulfates, chlorides, residual chlorine, and total hardness of the water, were conducted over a period of around three months to create a reliable database that will be used to evaluate the network efficiency and build prediction models that can estimate the amount of free chlorine remaining in the network and investigate how it relates to other parameters.

3.3.3 Methods of laboratory testing

The physical characteristics and pH test were examined at the Sanitary Laboratory in the Civil Engineering College at the University of Karbala, while the chemical properties were assessed at the Public Health Laboratory in Karbala. except the alkalinity test was evaluated in the Chemistry Laboratory at the Karbala Environmental Directorate. As for the residual chlorine, it was examined on-site.

3.3.3.1 Free RC test

This test was executed at the site by the researcher. According to the standard DPD colorimetric protocol for the quantification of free residual chlorine, as outlined in the Standard Methods for the Examination of Water and Wastewater (APHA, 2017). and this test was achieved through five steps: select a bottle sample, and the bottle was sterilized with distilled water before use, then the bottle was filled to the top to ensure proper sample representation. After that, during water collection, the free RC test was immediately conducted using a free RC testing tool through the color

comparison method. Water is added to the part designated for chlorine testing, then Lovibond DPD No.1 tablets are added, one piece to show the color, as shown in the Figure (3-8).



Figure (3-8): Residual Chlorine tools and test.

3.3.3.2 PH test

The pH of water indicates the acidity or alkalinity of a fluid solution. The pH scale regularly ranges from 0 to 14. The scale is logarithmic, not linear. In other words, a solution with a pH of 6 is ten times more acidic than a solution with a pH of 7. Pure water is considered neutral, possessing a pH of 7. Water with a pH below 7.0 is classified as acidic, and water with a pH above 7.0 is classified as alkaline (**Khor et al., 2024**). The test occurs in the laboratory with a Multiparameter water quality analyzer and its model is M99720 after the collection of samples. The pH measurement was conducted in accordance with the standard electrometric method (**APHA, 2017**). The instrument was calibrated with standard buffer solutions at pH levels of 4.0, 7.0, and 10.0 to guarantee precision. The electrode is washed with distilled water prior to use, subsequently immersed in water, and the reading is

recorded once the value stabilizes. The procedure is reiterated for each sample, as shown in Figure (3-9).



Figure (3-9): PH test.

3.3.3.3 Electrical conductivity (EC) and Dissolved solids testing (TDS) and Temperature (Temp) test

TDS quantifies the total concentration of all inorganic and organic substances as well as salts present in water (**Rafiqul Islam, 2016**). The EC and TDS tests are similar to the pH test and utilize the same equipment, with the exception that the pH electrode is replaced with one specifically designed for TDS and EC measurements. The tests were performed according to with the Standard Electrometric Method (**APHA, 2017**). Readings are obtained once the value stabilizes, and the indicated temperature value is then recorded, as shown in Figure (3-10).



Figure (3-10): TDS and EC test.

3.3.3.4 Turbidity test

Turbidity is an essential physical characteristic of drinkable water quality. This physical characteristic leads to light being absorbed and scattered by particles instead of passing through a water sample in straight lines. Suspended impurities that hinder water clarity and lead to turbidity include silt, clay, fine particles that represent organic and inorganic material, soluble colored organic molecules, plankton, and additional microscopic creatures (**Misaghi et al., 2017**). Turbidity is measured using a Turbidity meter model (H193703-11). Based on Hanna Instruments (**Instruments, 2025**), the device adheres to international turbidity measuring standards as specified on its product page, complying with ISO standards, and measurement followed the Standard Nephelometric Method in accordance with **APHA (2017)**. Turbidity is a physical property that results in both the absorption and scattering of light, not its passage. Since the quantity of scattered light

increases with increased turbidity. The scattering of light produced by a solution is predominantly attributed to the suspended solids contained within it. It is measured in formazin nephelometric units (FNU), as shown in Figure (3-11).



Figure (3-11): Turbidity meter.

Before doing a turbidity test, the water bottle is shaken to avoid particulate deposition. The device's tube is washed with distilled water. Then, the sample water is placed on the marked line in the tube, ensuring the tube's surface is dry to prevent ray scattering. Then, it is placed inside the device and read. The other samples are examined in the same manner.

3.3.3.5 Chloride test (Cl^-)

To perform the chloride test was achieved by the Silver Nitrate Titration Method (APHA, 2017), a conventional technique for chloride analysis in water, 50 ml of the sample of water was placed inside a conical container with a capacity of 250 ml. then two drops of potassium chromate reagent were

introduced to the sample. The prepared solution was titrated with silver nitrate (0.014) until the color change from yellow to a cloudy brown occurred, and the volume of the added solution is recorded as shown in Figure (3-12). The subsequent equation was employed to determine the chloride concentration.

$$\text{Cl(mg/L)} = \frac{\text{A} * \text{N} * 35.45 * 1000}{\text{Volume of sample}} \quad (3-2)$$

A represents the volume of titration solution (silver nitrate) Sample in (ml), and N represents the normality of AgNO_3 (Baird et al., 2017) .



Figure (3-12): Chloride test

3.3.3.6 Sulfate test (SO_4^{2-})

Sulfate was measured by employing a standard method for sulfate analysis in water, the Barium Sulfate Precipitation Method (APHA, 2017). The sample was prepared in 50 milliliters, and a standard calibration solution was prepared to calibrate the spectrophotometer device. Next, the water sample was mixed with 0.05 g barium chloride (BaCl_2). It was measured by a sensitive scale about one minute, acting as an additive that interacts with

sulfates. The compound was barium sulfate (BaSO_4). The sample appears as a white precipitate, subsequently deposited in the device's tube, from which the reading was directly obtained, as shown in Figure (3-13).



Figure (3-13): Sulphate test

3.3.3.7 Total Hardness test (TH)

The EDTA Titrimetric Method, a common technique for determining water hardness, was used to calculate TH (**APHA, 2017**). Throughout this procedure, 50 ml of the water sample was mixed with one to two ml of a pre-prepared buffer solution to elevate the pH to 10. Following that, 0.1 grams of the indicator Eriochrome Black T was introduced to impart a wine-red color to the water. The sample continues to be titrated with the EDTA solution until a colorimetric transition to clear blue, as shown in Figure (3-14), after which

the volume of EDTA utilized was documented. Then, equation (3-3) was used to determine it.

$$\text{TH as mg CaCO}_3/\text{L} \quad (3-3)$$

$$= \frac{A * B * 1000}{\text{Volume of sample}}$$

A represents the titration for the sample (ml), and B represents mg CaCO₃ equal to 1 ml EDTA titrant (Baird et al., 2017) .



Figure (3-14): Total Hardness test.

3.3.3.8 Calcium test (Ca⁺²)

The EDTA Titrimetric Method for Calcium Hardness was used to determine the calcium concentration in compliance with the Standard Methods (APHA, 2017). The calcium test was performed in the laboratory by preparing 50 ml of water, which was combined with two milliliters of NaOH,

producing a pH of 12. Incorporating just a small amount of the murexide indicator results in a faint red color. The sample undergoes titration with EDTA Na while being stirred continuously until the faint red color transitions to purple, remaining unchanged with the addition of one or two further drops of the titration solution Figure (3-15).

$$\text{Ca}^{++} \left(\frac{\text{mg}}{\text{L}} \right) = \frac{\text{A} * \text{B} * 400.8}{\text{Volume}} \quad (3-4)$$

A represents milliliters of titrant for the solution, B signifies milligrams of CaCO_3 that equal 1 milliliter of EDTA titration for the calcium indicator threshold (Baird et al., 2017) .



Figure (3-15): Calcium test

3.3.3.9 Alkalinity test (Alk)

In compliance with APHA APHA (2017), the alkalinity test was conducted using the Standard Titrimetric Method. Conducting an alkalinity test, 50 milliliters of the sample were put in a flask, followed by the addition of two drops of methyl orange indicator, and it was subsequently titrated with

a calibrated 0.02 N H₂SO₄ solution. The experiment is illustrated in the Figure (3-16). The following equation is applied.

$$\text{Alk. Caco3} = \frac{V * N * 50000}{\text{Volume of sample}} \quad (3-5)$$

Where V represents titration the sample (ml) and N is the Normality of H₂SO₄ (Baird et al., 2017).



Figure (3-16): Alkalinity test

3.4 Programming Work

3.4.1 Cleaning data

After completion of all tests, the data was organized into an Excel sheet for simplified management, followed by the execution of the data cleaning process. This means that all the data was filtered by removing out layers by using box plots and substituting the missing values by using linear interpolation, and data mean of the data. A boxplot is easy and understandable that displays the median, the first and third quartiles, and any outliers within a sample. After detecting the median and quartiles of the data, compute the

Interquartile Range IQR, which refers to the difference between the third and first quartiles, represented by Q3 and Q1. It was considered an important value for finding outliers using the following equations:

$$Q3 + 1.5IQR \quad (3-6)$$

$$Q1 - 1.5IQR \quad (3-7)$$

The Box plot is illustrated in Figure (3-17) **(Perez &Tah, 2020)**.

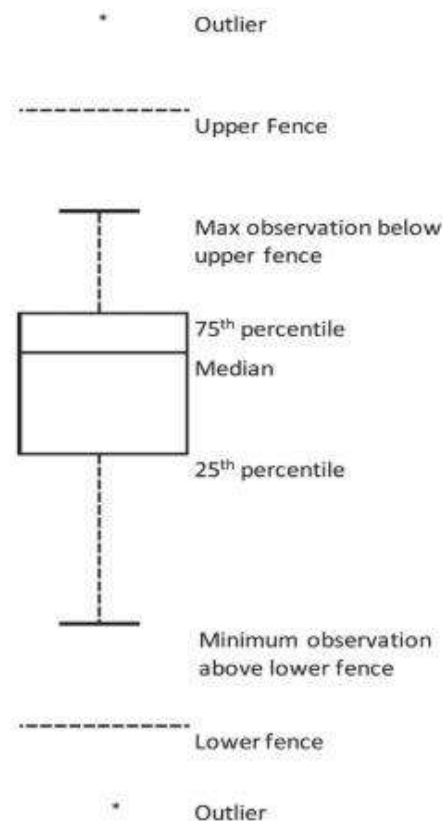


Figure (3-17): Box plot (Perez and Tah, 2020).

3.4.2 The software used

3.4.2.1 Python 3

Python is the high level, a broad-application programming language created in 1991. It was utilized across various platforms, including desktop

applications, server-side scripting, ML, web development, and data analysis (**Python Software Foundation, 2025**) Python is the leading language in the field of data analysis, and the growing importance of data has rendered proficiency in Python an essential skill to attain. As well as a broad variety of libraries, it renders it an ideal instrument for tasks ranging from data analysis as well as ML to graphical representation and developing websites. Beginning to work with Python needs setting up a suitable environment (**Kumar & Saharia, 2024**).

3.4.2.2 Jupyter Notebook

It is an interactive environment that integrates the execution of code, complex text, mathematics, visualizations, and multimedia (**Swamynathan, 2019**). It is considered an open-source dependency on a web-based system that has become a favored tool of data scientist for academics, and scientists. The user interface is easy and strong, allowing users to generate and publish files including live code, math formulas, data visualizations, and informative text. A principal attribute of a Jupyter Notebook is its versatility in supporting over 40 programming languages, such as Python, R, Julia, and Scala, rendering that make it adaptable to various computational needs and projects. It allows users to change between different languages within the notebook and rendering it extraordinary (**Kumar & Saharia, 2024**).

3.4.2.3 Anaconda

The Anaconda package manager provides a freely accessible distribution of the Python and R programming languages for computer science, with the goal of facilitating package management. It is a combination of more than 1,500 packages in calculations, data visualization, analysis, ML, and includes essential libraries like Pandas, NumPy, Matplotlib and others. Furthermore, it has the package manager, Conda, which facilitates

environment management, consequently simplifying the setup and maintenance of various software package versions (**Kumar & Saharia, 2024**). Anaconda, a platform that efficiently integrates Python with numerous data science libraries (**Kumar & Saharia, 2024**).

3.4.2.4 SPSS

The IBM SPSS Statistics program is a smart and easy to use also strong statistics software by every standard. SPSS software from the IBM Company. SPSS can do virtually any statistical analysis utilized in the social sciences, commercial field, and several scientific fields (**Darren George, 2016**). The tools offered by the program's available version were utilized to perform the boxplot approach in this study.

3.4.3 Input Variables

The variables are input through three different scenarios.

In the first scenario

Input X: Represents all inputs for Group 1 and Group 2, including distance, time, and date, which are referred to as (dataset 1).

Outputs Y: represents the RC.

In the second scenario

Input X: Represents only the Group 1 with distance, time, and date, which are referred to as (dataset 2).

Outputs Y: represents the RC.

In the third scenario

Input X: Represents only the Group 2 with distance, time, and date, which are referred to as (dataset 3).

Outputs Y: represents the RC.

3.4.5 The Analysis of Data

3.4.5.1 Preprocessing data

The Pandas package in Python has been utilized to arrange the structure of data columns, facilitating efficient analysis and development for ML models. To facilitate the handling data, the time column was converted into minutes since midnight, while the date column was broken down into individual year, month, and day features. The lack of historical data within the DWDN and the limited monitoring of that data to just free residual chlorine testing leads to insufficient data for adequately training the models. Consequently, the data was artificially augmented by the augmentation technique in Python using the numpy library. The data augmentation, is a method that produces data and enhances the efficacy of the ML model **(Maharana, 2022)**. where this method represents a strategy applied to the data to solve the issue of insufficient data. It is based on the concept that extra data can be obtained from the original data via augmentations. The augmentation, artificially increases the training quantity through oversampling or even warping data. The benefit of augmented data is reducing overfitting by transforming restricted datasets to exhibit the attributes of large amounts of data **(Shorten, 2019)**. The augmentation instituted to maintain patterns during the training phase of the model, provided by numerical data, is through a process wherein the original data was replicated twice, and random noise was introduced to the duplicated data at a rate of 5%. In other words, the noise was added to all parameters. Except for time, date, and distance, denoting the noise factor, to closely approximate the actual data.

This method was instituted to prevent overfitting and also to maintain patterns during the training phase of the model. Noise is characterized as meaningless information that contributes to data distortion. ML systems must be responsible for noise in synthetic data to identify patterns within it. This minimizes overfitting and improves model performance at the output, considering the impossibility of acquiring noise-free data. Commonly employed methods involve Gaussian noise and impulsive noise. Also, the noise is of the type salt-and-pepper, and quantization noise (**Somanna, 2023**). Gaussian noise was added to the dataset using the NumPy library in Python.

3.4.5.2 Determine T-Test and P-Value

The tests were performed to verify that the adjustments in the data, after the introduction of noise, did not yield statistically significant variations that might affect the data's properties. using Python's Scipy library, the t-test and p-value calculations were performed for all variables except time, date, and distance. for t-test has been calculated by performing the code employing the `t-test_ind_from_state` function to comply with Welch's equation, together with degrees of freedom according to the Welch-Satterthwaite equation (**Ruxton, 2006**), directly from the library, as shown in the equation:

$$t - test = \frac{\mu 1 - \mu 2}{\sqrt{\left(\frac{s1^2}{n1}\right) + \left(\frac{s2^2}{n2}\right)}} \quad (3-8)$$

Where: $\mu 1$ and $\mu 2$ represent the means of the original and augmented data, respectively. Along with $s1$ and $s2$ denote the standard deviations of the original and augmented datasets, respectively, where $n1$ and $n2$ represent the sample sizes for the original and augmented datasets, respectively. The P-value was calculated using the equation (3-9).

$$P=2P(T>|t_o|) \quad (3-9)$$

Where: t_o is the absolute value of the computed t-value, and T is a random value **(Ahad & Yahaya, 2014)**. The P-value results are subsequently compared to 0.05. If the resultant value exceeds 0.05, it means that the differences are not significant, and vice versa **(Kwak, 2023)**.

3.4.5.3 Splitting data

The data has been divided into a training set to instruct the models on the structural patterns and a test set to assess the models' performance on data different than the training set. The data was partitioned using the popular Python library Scikit-learn. This library applies the principle of random division to guarantee an equitable distribution of the data. Utilizing the train-test split function, 70% was allocated for training and 30% for testing, a ratio selected due to its common usage.

3.4.5.4 Data Normalization

The Standard Scaler is utilized via the scikit-learn module in Python to ensure that the mean equals zero while the standard deviation equals one. Where it operates on the training along with testing datasets. The purpose is to eliminate substantial discrepancies in measuring units or ranges among variables. The equation (3-10) was used for calculating data normalization **(scikit-learn developers, 2023)**.

$$Z = \frac{x - \mu}{\sigma} \quad (3-10)$$

Where x is the original value, and μ = mean of the sample, and σ = standard deviation.

3.4.5.5 The hyperparameter optimization for algorithms

A main objective and difficulty with the ML process is enhancing the performance results, located in data patterns and observed information. To succeed at this purpose, approximately each of the ML algorithms possesses an individual set of parameters that must be evaluated from the dataset to optimize the performance result. The optimal method for selecting effective hyperparameters depends on exhaustive trial and error with every possible parameter value combination. Scikit-learn offers functions of the GridSearchCV as well as RandomSearchCV to enable an automated and reproducible method for hyperparameter optimization. The principle of how GridSearchCV works is that a set of adjustable parameter values can be identified. Then, utilizing GridSearchCV, models are constructed for all possible combinations from a predefined list of hyperparameters, with the optimal combination selected based on cross-validation results (Swamynathan, 2019). The GridSearchCV approach was employed in this study, with proven values for the hyperparameters essential to each algorithm. This strategy involves testing numerous combinations until the one that produces optimal model performance is identified.

3.4.5.6 Building AI models

This study included five primary algorithms: Random Forest (RF), XGBoost, General Regression Neural Network (GRNN) for ML, Transformer, and Multilayer Perceptron (MLP) for ANN. These algorithms were chosen to make a model for their ability to forecast residual-free chlorine levels in DW within the WDS due to ability to handle diverse data. Optimization of algorithm hyperparameters. frequently, selecting the most effective design of a ML algorithm involves a process of experimentation. The

model hyperparameters regulate the learning operation. Several algorithms possess fixed hyperparameters, while others necessitate extensive fine-tuning (**M. I. Hossen Abir, 2024**).

A. GRNN

This algorithm was chosen because of its proven superiority in previous studies, and it is efficient in predicting disinfection levels. Concerning the GRNN algorithm, there exists no specific library; instead, the code was developed directly within the Python Jupyter Notebook environment. In GRNN models, the spread factor is crucial for developing a strong ANN model of regression characterized by great forecast precision as well as reliability (**Wadkar & Kote, 2017**). In this study, the spread factor was represented by a (sigma) or (smoothing parameter) σ , which regulates the Gaussian kernel's width. It calculates the speed at which the Gaussian function declines with increasing separation between training samples and a new input. Large σ results in wide kernels (many training points affect the output), while small σ results in narrow kernel widths (only very close training points have influence) (**Specht, 1991**). Which determined the optimal value for it through trial and error.

B. Multi-Layer Perceptron (MLP)

It is a multilayer neural network constructed with the Tensor Flow library in the domain of neural networks. GridSearchCV selected the optimal hidden layer size, neuron count, activation function and regulation parameter (alpha). The Solver denotes an optimization method calibrated to Adam, a symbol for "Adaptive Moment Estimation" (**Saman et al. , 2021**). Adam is a method designed for model optimization. It uses first-order gradient descent and is employed in neural network training. It is characterized by its rapidity and

effectiveness in operation, while requiring less memory usage. And it is also appropriate for challenges regarding the amount of data or feature numbers. Adam's approach is more efficient than other random optimization techniques **(Kingma & Ba, 2015)**.

C. Random Forest (RF)

RF Regression is an approach that averages the outputs of decision trees to achieve higher quality results **(Onyutha, 2022)**. RF is a group decision tree approach that utilizes the results from many equivalent weak decision trees to generate forecasts. The weak tree arbitrary partitions a minor segment of the training data and input values, concluding in an overall RF forecast that represents the average of the weak trees **(Kyritsakas et al., 2023)**. The RF model was constructed utilizing the Python Scikit-learn package. To adjust the hyperparameters in the RF algorithm, the GridSearchCV method from the Scikit-Learn package was utilized. The hyperparameters that were introduced are:

n-estimator: represents the number of weak trees that are included in the forest.

Max depth: reflects the tree's maximum depth.

Min sample split: the smallest number of samples taken to separate each node in the tree.

Min samples leaf: the smallest number of samples taken in each leaf node.

D. XGBoost

Tree boosting is an exceptionally efficient and extensively employed ML technique. XGBoost is extensively utilized by data researchers to get superior

outcomes for multiple ML problems (**Chen & Guestrin, 2016**). The hyperparameters were determined using the GridSearchCV technique, and the hyperparameters used were

n-estimator: number of trees.

Max depth: the maximum depth for the tree ranged from 5 to 30.

Learning rate: Learning rate of the model during each iteration ranged from 0.005 to 0.2.

Subsample: Proportion of samples utilized for training in each tree.

Colsample by tree: The proportion of features utilized in each tree ranged from 0.8 to 1 (**Ris *et al.*, 2019**).

E. Transformer

A Transformer model suitable for regression tasks and for time series prediction has been created using the PyTorch library to check the model's capacity to forecast RC. This model follows the original design proposed by (**Vaswani *et al.*, 2017**) in structure; however, it eliminates a decoder as it predicts a singular value, RC. Where this model involves layers as illustrated in the Figure (3-19). The encoder consists of an input layer that transforms incoming data into an internal numerical format understandable by the model, employing dimensions d model through a fully connected network. Then, a positional encoding layer is employed to encode the time sequence of the data. As for the third layer, it consists of multiple layers, each of which contains a self-attention layer that enables the model to concentrate on all sequences. Also, the feedforward network layer, which represents a basic neural network, processes the output from self-attention. Subsequently, the normalization

layer is crucial for ensuring the stability of the training process (**Wu et al., 2020**). The resultant sequence is transformed into a singular vector to forecast the RC value.

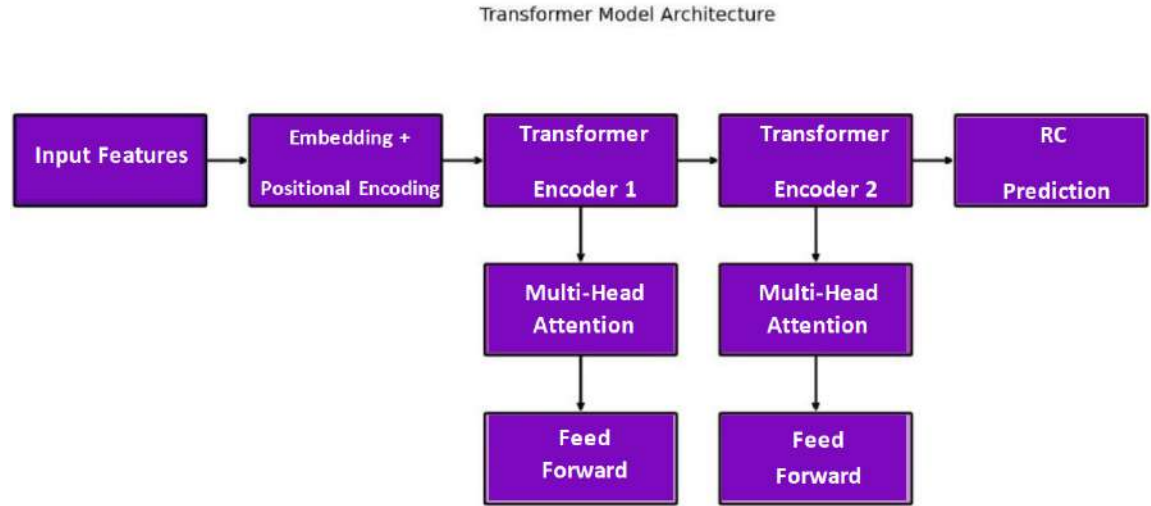


Figure (3-19): Transformer Model Architecture.

Due to the model's dependence on the sequence, a window size of 8 has been identified. And for the critical hyperparameters influencing the model's performance, involving the number of layers (d-model), the number of self-attention heads (num Head), and the learning rate, have been discovered using the GridSearchCV method.

3.4.5.7 Performance metrics

The effectiveness of the optimal algorithm is evaluated and improved based on its capacity to predict both training and new data, which is essential for obtaining maximum accuracy (**Engelbrecht, 2007**). Checking the coefficient of determination (R^2) is a prevalent way for evaluating actual outcomes against ideal predictions, with $R^2=1$ denoting an accuracy level of 100%. Conversely, the mean square error (MSE) quantifies the mean error in forecasting across the entire dataset (**Wadkar & Kote, 2017**). The models

were evaluated using performance measures based on the test data, which constitutes 30% of the total. From the scikit-learn library, the MSE was determined using the `mean_squared_error` function, as well as the MAE using the `mean_absolute_error` function, while R^2 was determined using the `r2_score` function. Where The range of coefficient of determination R^2 from 0% to 100% or (0-1) (**Investopedia, 2009**). And the worst value for MAE, MAPE, MSE, and RMSE is $+\infty$, and the greatest value is 0 (**Chicco et al., 2021**).

$$R^2 = 1 - \frac{[\sum_{i=1}^m (X_i - y_i)^2]}{[\sum_{i=1}^m (\bar{Y} - y_i)^2]} \quad (3-11)$$

$$MSE = \frac{1}{m} \sum_{i=1}^m (X_i - y_i)^2 \quad (3-12)$$

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (X_i - y_i)^2} \quad (3-13)$$

$$MAE = \frac{1}{m} \sum_{i=1}^m |X_i - y_i| \quad (3-14)$$

m represent the number of data and X_i represent what was forecast and Y_i is the actual value , $\bar{Y}$ = mean of the real value (**Chicco et al., 2021**).

3.5 Summary

This chapter presents data collected over a three-month period from laboratory and fieldwork on two pipes in the Al Hussain neighborhood DWDN. The data was modified, cleaned, and outliers removed using the box plot method. It was then integrated into AI models. Subsequently, it had been integrated into a Python working environment, where preprocessing involved formatting the columns to conform to Python standards with features

designated as inputs and outputs. Three scenarios were selected for analysis. The Scikit-learn library was used to partition the data, and a standard scaler was used to integrate into models like GRNN, XGBoost, RF, MLP, and Transformer. The chapter establishes the structure and approach for the results presented in the fourth chapter.

Chapter Four: Results and Discussion

4.1 Introduction

This chapter highlights the data analysis for the first and second directions, comparing it with, the WHO (2017) standards and Iraqi Specifications No.417 (2009). The principal outcomes obtained by training the AI models in both directions were examined, along with the key illustrations that emphasize the results for the values based on the models that were evaluated throughout all scenarios that were developed and enhanced. Additionally, the models were compared, the main reasons for this disparity were discussed, and the best model was identified.

4.2 Data Analysis

4.2.1 PH

As per the WHO guidelines (**WHO, 2017**), and to the Iraqi Standard Specification for DW 417 (2009), the pH range is between 6.5 and 8.5. Figure (4-1(A)) illustrates that the pH values for all points in the direct 1 ranged from 6.87 to 7.95.

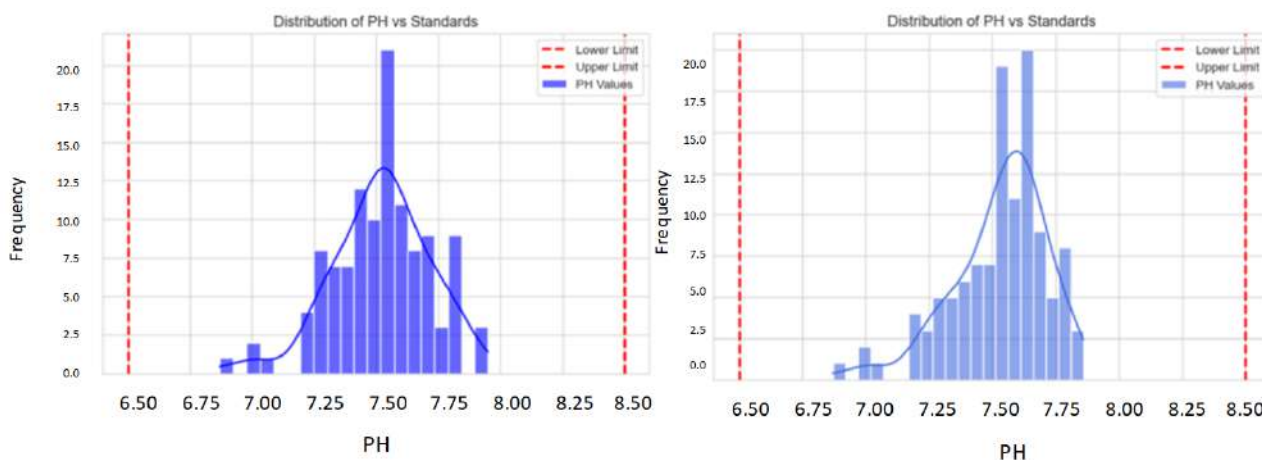


Figure (4-1): Distribution of PH and Standards (A) direct 1 (B) direct2.

Also Figure (4-1(B)) represent PH distribution in direct two with range (6.87,7.86). This means that the pH inside the Plant and the network is considered acceptable and in line with those guidelines.

4.2.2 Turbidity

Figure (4-2(A)) illustrates that the turbidity in the direct 1 ranged (2 to15.4) at project site P0 exceeded the allowable limit in 19 readings, representing about 65.5% of the data at point P0. However, Figure (4-2(B)) illustrates the distribution of turbidity measurements in P1, with over 12 readings outside the thresholds, representing roughly 41.37%, and for P2 in Figure 3C, demonstrating that 10 readings exceed the thresholds, approximately 34.48%.

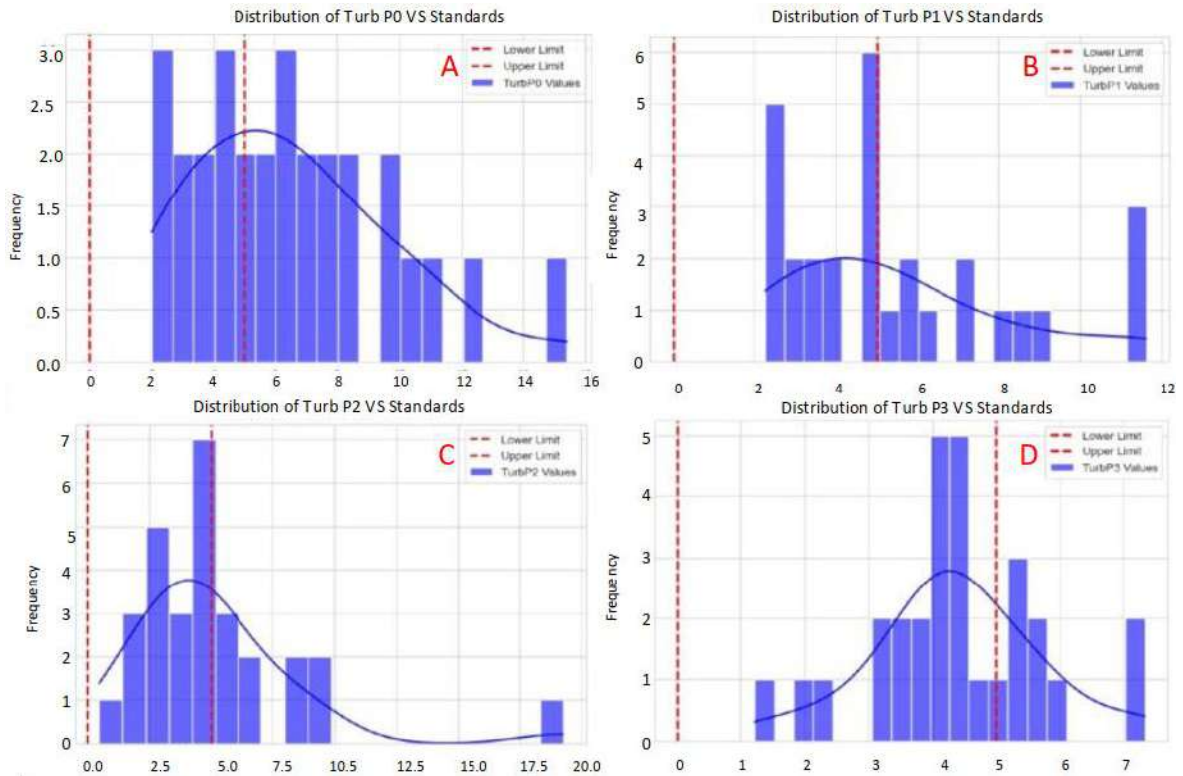


Figure (4-2) :Distribution of Turbidity in direct 1 for (A) P0 (B) P1 (C) P2 (D) P3.

As for Figure (4-2(D)), it is for point P3, where 9 readings exceeded the turbidity thresholds by about 31%. The turbidity in the direct 2 showed in Figure 4-3, The turbidity readings at P0 surpass thresholds by 18 readings, approximately 62% Figure (4-3(A)). As for P1 Figure (4-3(B)), it exceeded by 13 readings, which was equivalent to (44.8%). For P2, it exceeded by 20 readings, representing 68.9% Figure (4-3(C)). P3 exceeded expectations by 21 readings, representing 72.4% in Figure (4-3(D)).

The evaluation of turbidity data for both directions found that the values surpass the permissible limits set by the WHO and the Iraqi guidelines for DW turbidity, which should not exceed 5 NTU. In that first direction, the turbidity exceeding rate in the plant starts at 65.5%, decreases to 41.37% at points P1, 34.48% at points P2, and finally reaches 31% at points P3.

While in the second direction, turbidity began at a high level of 62% and subsequently dropped to 44.8% at P1. Whereas the percentage of measurements over the limits in P2 and P3 increased to 68.9% and 72.4%, respectively, indicating larger percentages than those recorded within the plant point P0. The data analysis exposed the general situation in both directions, demonstrating that the turbidity level begins high at the source. The elevation of turbidity over the permissible threshold in treated water is usually explained by the insufficiency of filtration and sedimentation processes in removing the floc created by the introduction of aluminum sulfate (alum) in the sedimentation tanks utilized for coagulating and settling suspended particles **(Al-Saadi, 2013, as cited in Al-Nasrawi, 2014).**

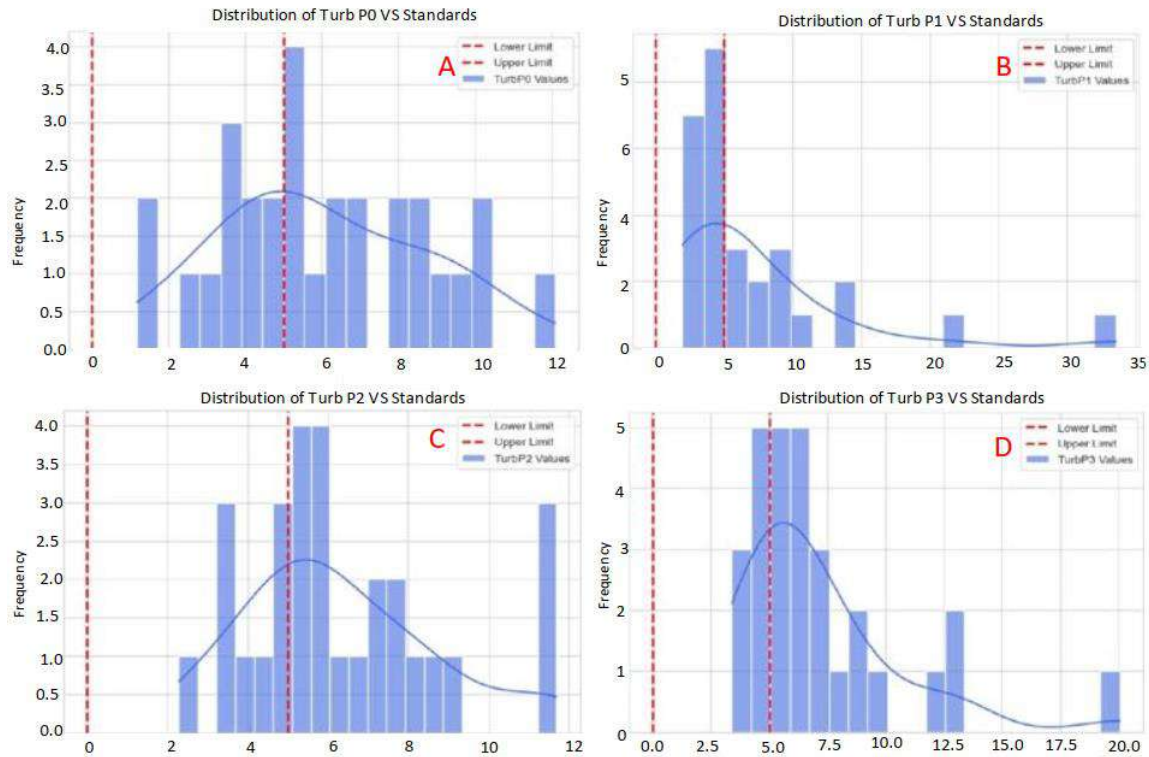


Figure (4-3): Distribution of Turbidity in direct 2 for (A) P0 (B) P1 (C) P2 (D) P3.

The reduction in the ratio of readings that exceed the limitations with increasing distance in the first direction until the end point, as well as the first point in the second direction, might be explained as a result of the deposition or reactions of organic particles during the flow of water through the pipes. While the rise in turbidity at locations P2 and P3 may be attributed to reasons to the turbidity in water pipes may increase due to the corrosion of metals in the pipe substance, such as iron as well as manganese, or as a result of microbial growth that increases this corrosion (**Cepeda & Cepeda, 2006; Al-Yasari, 2012, as cited in Al-Nasrawi, 2014**). It complies with the (**Anne *et al.*, 2023**) research, which demonstrates a rise in turbidity inside the network due to Insufficient maintenance of networks, resulting in a collection of sludge within the system. It is also compatible with the research conducted by **Al Chalabi (2020)**, which evaluated the quality of the DW plant in Basra,

demonstrating that turbidity levels are higher than allowed limits due to the plant's inefficiency. The study confirmed that the units, including sedimentation, filtration, and disinfection tanks, were functioning ineffectively, particularly when the facility treated low-quality raw water. Since turbidity is considered to have exceeded acceptable thresholds at specific points, this can subsequently influence the efficacy of sterilization when turbidity is above the permitted threshold. Therefore, this reduces the effectiveness of disinfectants in eliminating bacteria, making the water unsuitable for drinking (**Abed & Khudair, 2023**).

4.2.3 Residual Chlorine

Residual chlorine represents a crucial indicator of DW quality. According to the Iraqi standard No.417 (2009) requirements, the free RC concentration in DW shouldn't drop less than 0.3 mg/l at the final point of the water distribution network. As for the WHO, it suggested that the final concentration should be within the limit of 0.2 to 0.5 mg/l. RC levels in the distribution system may need to exceed the aesthetic target of 0.5 mg/L under specific conditions. In any case, the concentration of chlorine in DW provided to customers must consistently be below the WHO-recommended value of 5 mg/L (**WHO, 2017**). Figure (4-4) shows the histogram showing the distribution of RC in the direct 1 of the networks. Whereas Figure (4-4(A)) shows that the range of values from 2 to 6 mg/l at supply point P0 shows a significant rise in the RC percentage exceeding the allowed limit, seen in 6 readings, or 20.6% of all readings. For P1, which was governed by lower standards, as shown in Figure (4-4(B)), only one reading was below the limits, and most of the readings that were above the limits represent 96.5%, which was a point close to the Plant. In the case of P2, shown in Figure (4-4(C)), it

also had an overdose of (0.2–0.5 mg/l), with 27 readings being higher than the limits, or 93.1%. The final point, P3, achieved acceptable values at specific readings; nevertheless, at several readings, it exceeded the permitted limit by 19 readings, or 65.5% Figure (4-4(D)). This decrease is rational, considering its distance from the Plant. As for the direct two, Figure (4-5(A)) shows that at P0, there was an exceedance of the limits by 9 readings (which means 31%).

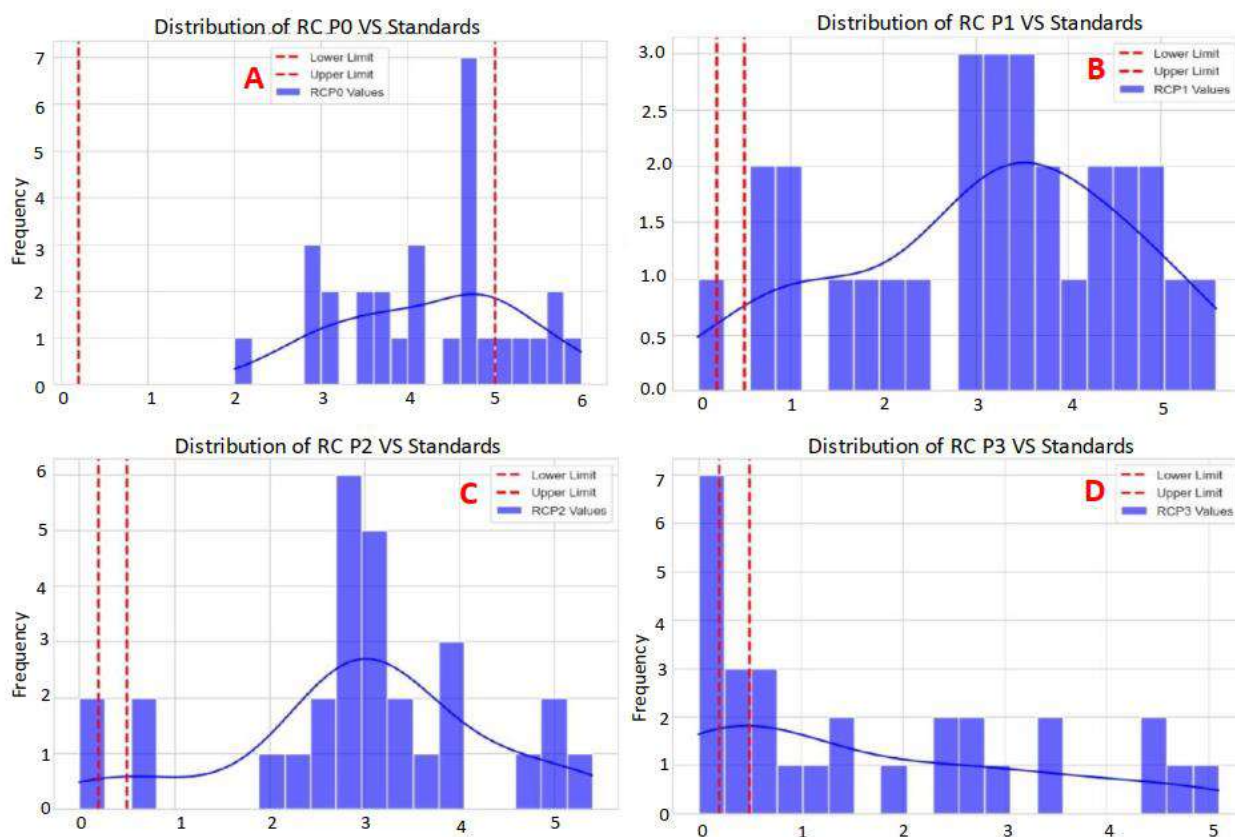


Figure (4-4): Distribution of RC in direct 1 for (A) P0 (B) P1 (C) P2 (D) P3.

For P1, the readings of the RC that exceeded the limits were 24 readings, which was equivalent to 82.75% as shown in Figure (4-5(B)). Also, for P2, it exceeded by 27 readings, which means 93% above the limits, Figure (4-5(C)). In addition, for P3, it was 13 readings, which means 44.8%, Figure (4-5(D)). The analysis of RC data in the direct one and two indicates an

excessive dosage at all points, which was unsuitable for the water to meet consumption standards within the established limitations. When the distance increased, the concentration of chlorine was reduced. An analysis of the RC data in both directions and at all sample points indicated a variance in RC level within the network. The variance in concentrations of RC is influenced by multiple factors, particularly the distance from the plant, the age of the network, degradation of the pipeline systems, and various technical considerations (Rammadhan et al., 2016).

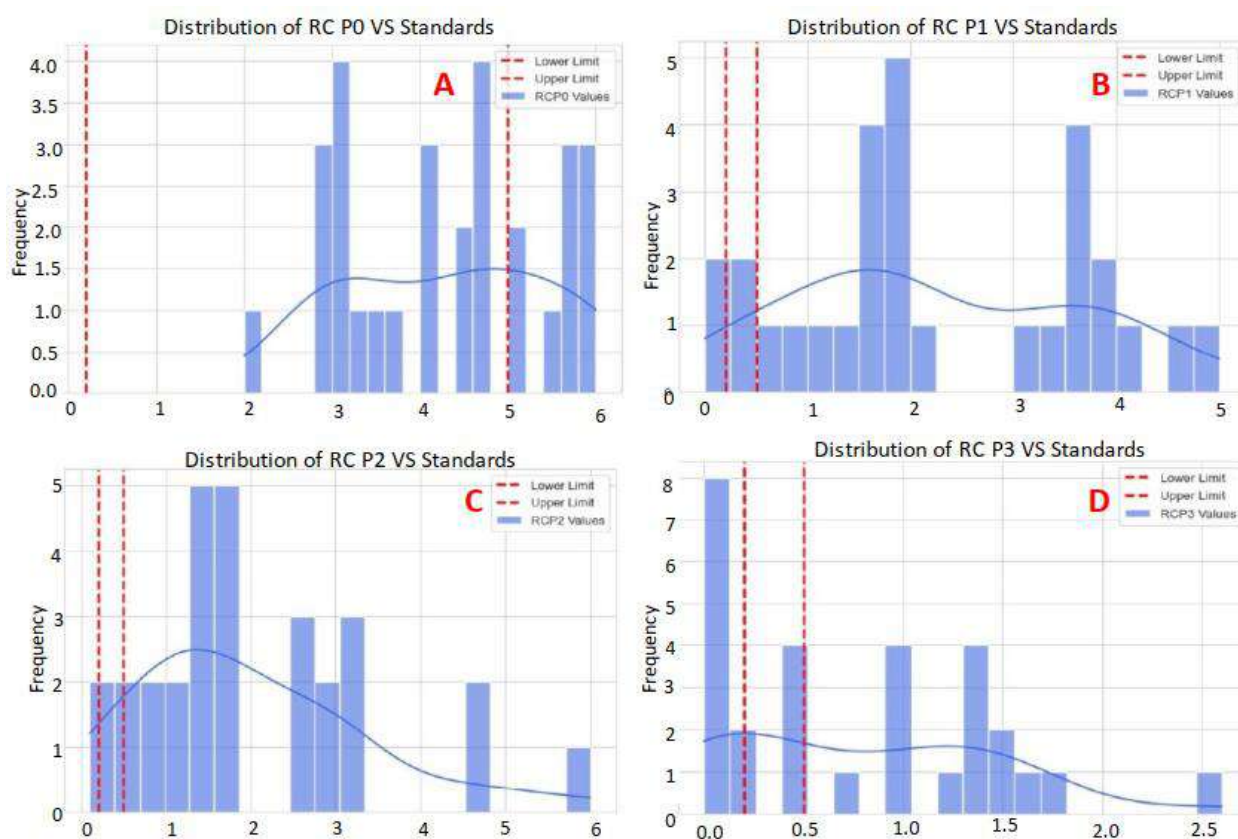


Figure (4-5): Distribution of RC in direct 2 for (A) P0 (B) P1 (C) P2 (D) P3.

Researchers have identified the disadvantage of elevated chlorine levels, where recent investigations have demonstrated that long exposure to chlorine may affect human health. Prolonged exposure to chlorinated water, whether through drinking or even swimming, may result in dental damage;

not only that, but there are indications that it may also be a carcinogen. Studies have indicated that exposure to chlorinated water during bathing may provide greater risks than consuming it through drinking (**Hameed et al., 2018**). The distribution network has consistently demonstrated an overdose, with both directions recording multiple readings over the threshold of 0.5 mg/l. Excessive RC concentrations may result in consumer complaints and deterioration of the piping system (**García-Ávila et al., 2021**). Where the dose control issues have become evident, and the disinfection process needs enhancement.

4.2.4 Total Dissolved Solid (TDS)

The TDS remained stable at all points in the first direction and did not exceed the maximum permissible limit specified, which is 1000 mg/l. Where the readings varied from 556 to 676 mg/l, as illustrated in Figure (4-6(A)). Also, in direct 2 in Figure (4-6(B)), the reading varied from 553 to 676 mg/l, This indicates that the TDS remains constant at around 600 mg/l.

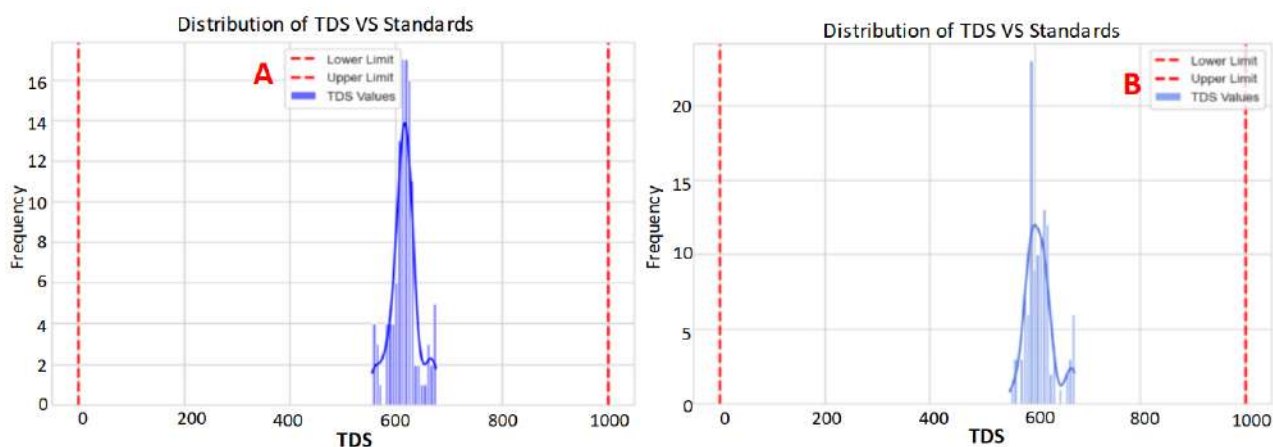


Figure (4-6): Distribution of TDS and Standards (A) direct 1 (B) direct2.

4.2.5 Total Hardness (TH)

The total hardness was within acceptable limits at all points, not exceeding the maximum threshold of 500 mg/l. Figure (4-7(A)) illustrates that the TH measurements for all points in the direct 1 varied from 290 to 450 mg/l, thus rendering them acceptable. The direct 2 varied from 280 to 430 mg/l Figure (4-7 (B)).

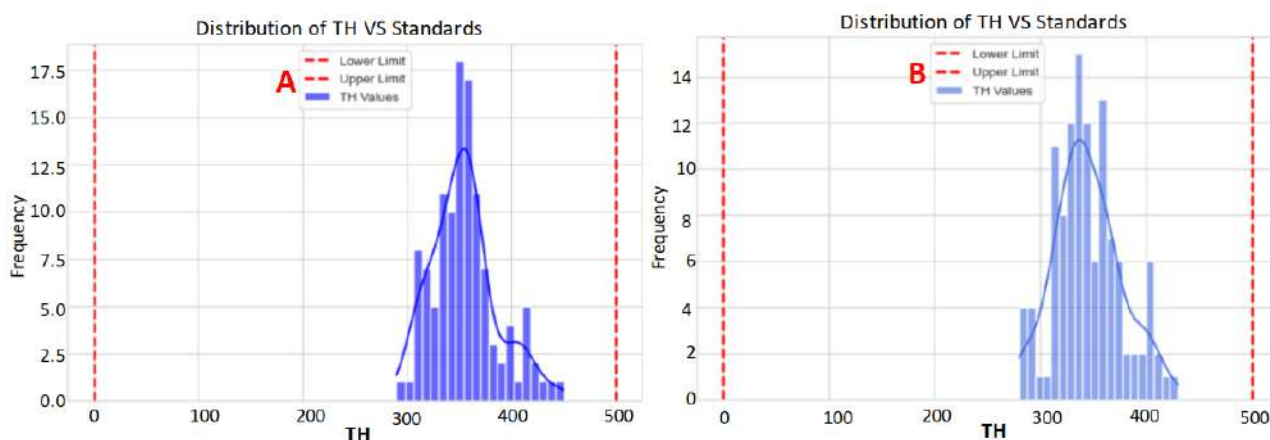


Figure (4-7): Distribution of TH and Standards (A) direct 1 (B) direct2.

4.2.6 Calcium (Ca^{+2})

Calcium levels were within accepted ranges, not above the maximum specified amount of 150 mg/l for direct 1, as illustrated in Figure (4-8(A)), where calcium values range from 55.2 to 112 mg/l.

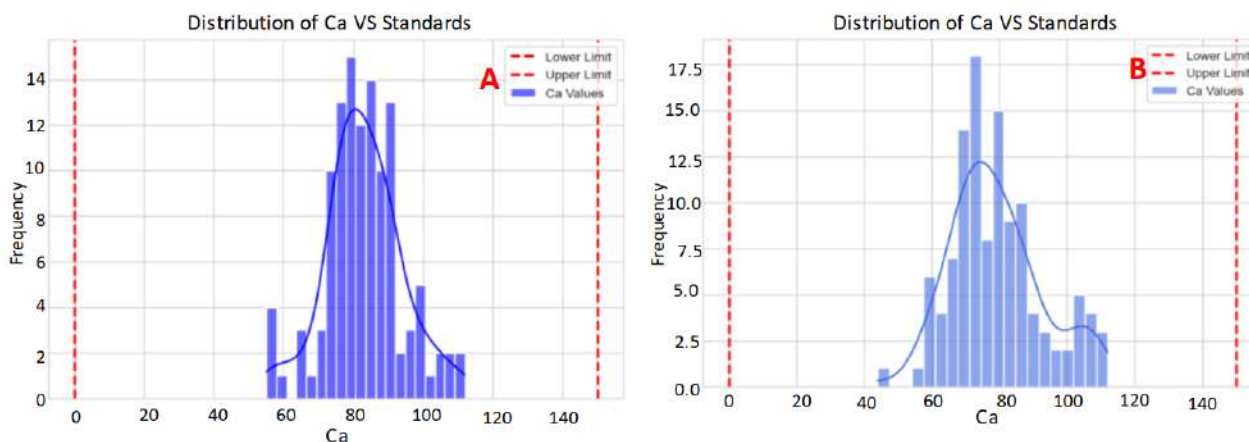


Figure (4-8): Distribution of Ca^{+2} and Standards (A) direct 1 (B) direct2.

Also, for direct 2, it ranges from 44 to 112 mg/l in Figure 4-8(B).

4.2.7 Chloride (Cl^-)

The distribution of chloride measurements in Figure 4-9(A) illustrates that the measurements for the points in the first direction ranged from 93 to 146 mg/l; as they remain below 350 mg/l, they were considered acceptable. The measurements for the second direction range from 92 to 145 mg/l, as depicted in Figure (4-9(B)).

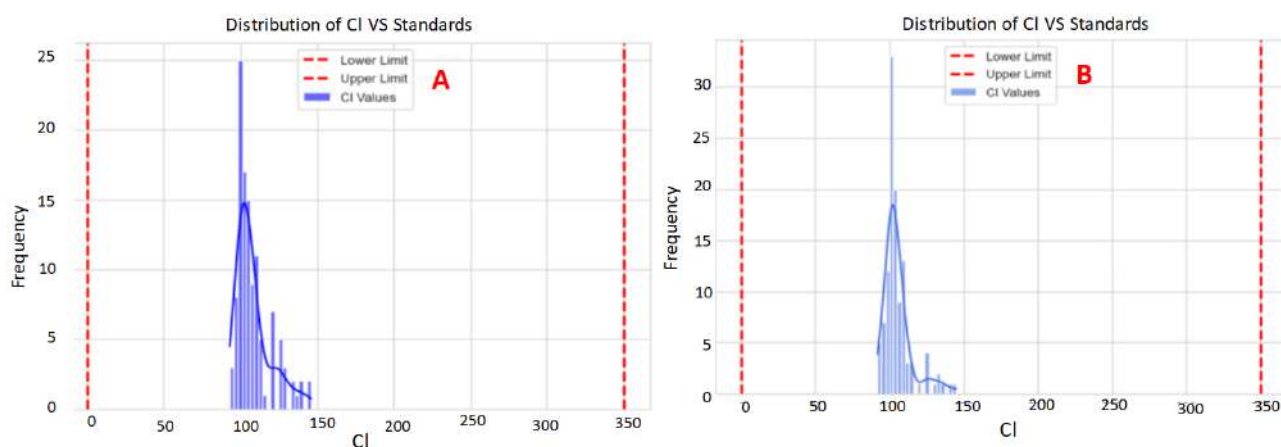


Figure (4-9): Distribution of Cl and Standards (A) direct 1 (B) direct2.

4.2.8 Sulfate (SO_4^{2-})

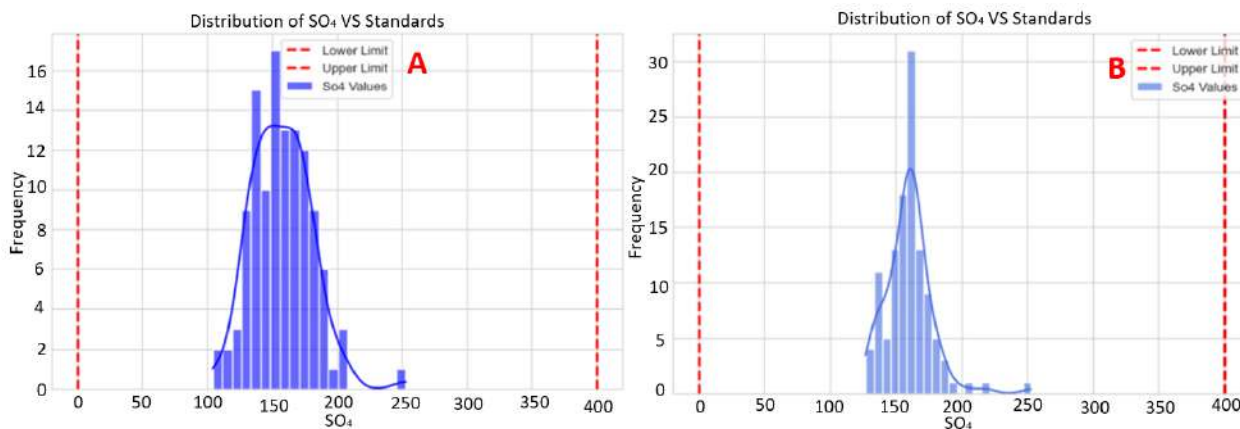


Figure (4-10): Distribution of SO_4^{2-} and Standards (A) direct 1 (B) direct2.

Figure (4-10(A)) illustrates the distribution of SO_4^{-2} readings in the first direction, ranging from 104 to 253 mg/l, demonstrating that they did not exceed 400 mg/l, the maximum permissible value within the standard limit. Consequently, it is permissible. The direct 2 ranged from 127 to 253 mg/l, Figure (4-10(B)).

4.2.9 Temperature

There are no specific limits on DW temperature, and the temperature of the water within the pipes was assumed to be constant. As for Figure (4-11(A) and (B)), both directions represent the temperature of samples recorded by the same device that measures the EC the in laboratory. It was included as an input variable due to its potential direct impact on other parameters and to provide an accurate representation of conditions, consequently enhancing the model's accuracy.

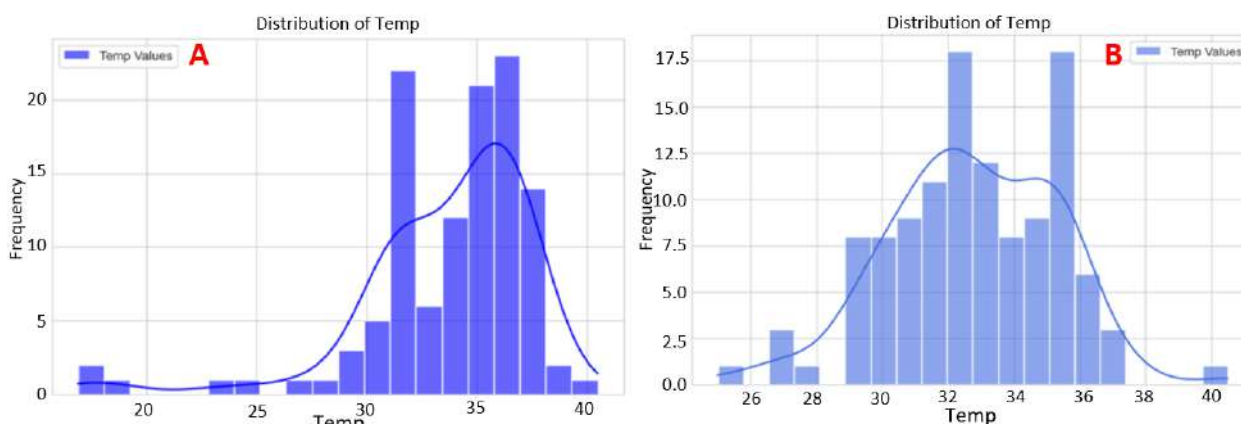


Figure (4-11): Distribution of Temp and Standards (A) direct 1 (B) direct2.

4.2.10 Electrical conductivity

The EC did not have limits, but they were added to the readings when creating the models due to their impact, starting with a range from 837 to 1056 $\mu\text{S}/\text{cm}$ for the first direction, Figure (4-12(A)). The second direction, it ranged

from 830 to 1050 $\mu\text{S}/\text{cm}$, Figure (4-12(B)), and it is very close in both directions in terms of range.

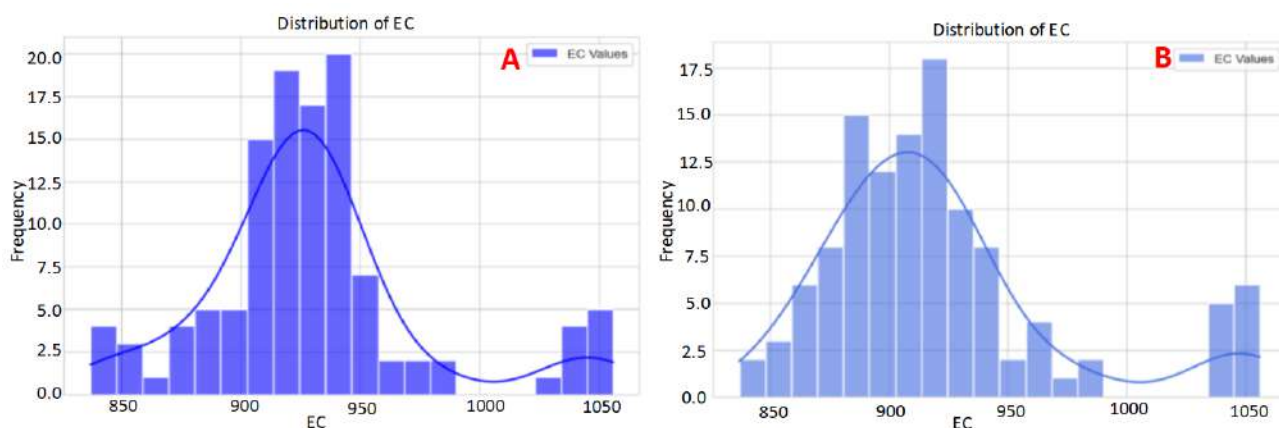


Figure (4-12): Distribution of EC and Standards (A) direct 1 (B) direct2.

4.2.11 Alkalinity

The standard does not include any limits concerning alkalinity. The interval in the first direction spread from 138 to 170, as illustrated in Figure (4-13(A)), with the peak frequency occurring at 145. In the second direction, the range extended from 110 to 185, with the peak frequency of values occurring at 145, as illustrated in Figure (4-13(B)).

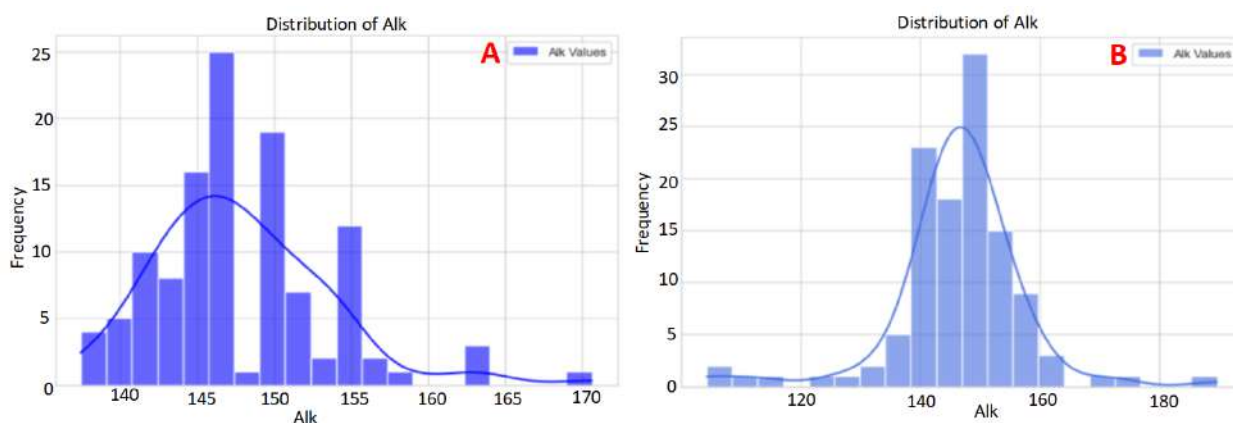


Figure (4-13): Distribution of Alk and Standards (A) direct 1 (B) direct2.

4.3 Data Augmentation

For direct one, Table (4-1) displays statistics for the original data, while Table (4-2) Augmentation data for direct 1.

Table (4-1) Original data for direct one

D1	COUNT	MEAN	STD	MIN	MAX	MEDIAN	RANGE
RC	116	3.00	1.65	0	6	3.05	6
Turb	116	5.35	2.91	0.46	19.26	4.79	18.8
PH	116	7.52	0.19	6.87	7.95	7.52	1.08
Temp	116	33.68	3.93	16.9	40.5	34.95	23.6
TDS	116	616.26	25.30	556	676	616	120
EC	116	929.50	45.89	837	1056	925	219
TH	116	354.21	31.33	290	450	352	160
Ca	116	83.17	11.07	55.2	112	83.05	56.8
Cl	116	108.30	11.62	93	146	105	53
SO ₄	116	157.09	23.09	104	253	156.25	149
Alk	116	147.77	5.61	137.46	170.64	146.94	33.18

Since the P-value exceeds 0.05 for all variables, this indicates an absence of significant differences between the means, demonstrating that the data remains consistent despite the introduction of noise. for the direct 2, Table (4-4) provides the statistics for the original data. While Table (4-5) exhibits the statistics for the augmented data. while the results from the P-value analysis, are presented in Table (4-6).

Table (4-2) Augmentation data for direct one

D1	COUNT	MEAN	STD	MIN	MAX	MEDIAN	RANGE
RC	348	3.00	1.64	0	6.00	3.09	6.00
Turb	348	5.36	2.91	0	19.45	4.74	19.45
PH	348	7.52	0.19	6.86	7.95	7.52	1.08
Temp	348	33.68	3.92	16.80	40.83	34.88	24.02
TDS	348	616.29	25.35	555.44	677.63	616	122.19
EC	348	929.16	45.77	837	1057.44	925	220.44
TH	348	354.25	31.30	288.34	450.28	352	161.93
Ca	348	83.17	11.07	54.74	112.69	82.77	57.95
Cl	348	108.28	11.59	92.30	146.15	104.67	53.84
SO ₄	348	157.15	23.05	102.54	254.63	156.50	152.08
Alk	348	147.77	5.61	137.29	170.70	146.94	33.41

Table (4-3): t-test and P value results for direct 1.

D1	T-test	P-value
RC	-0.003	0.99
Turb	-0.010	0.99
PH	0.006	0.99
Temp	0.006	0.99
TDS	-0.011	0.99
EC	0.070	0.94
TH	-0.010	0.99
Ca	-0.002	0.99
Cl	0.013	0.98
SO ₄	-0.023	0.98
Alk	0.0	1.0

Table (4-4): Original data direct 2.

D2	COUNT	MEAN	STD	MIN	MAX	MEDIAN	RANGE
RC	116	2.32	1.73	0	6	1.8	6
Turb	116	6.67	4.19	1.18	33.56	5.72	32.38
PH	116	7.53	0.18	6.87	7.86	7.55	0.99
Temp	116	32.72	2.57	25	40.5	32.65	15.5
TDS	116	609.29	26.09	553	676	606.75	123
EC	116	920.54	50.28	837	1056	912	219
TH	116	344.79	31.57	280	430	340.25	150
Ca	116	79.20	13.61	44	112	77.2	68
Cl	116	105.46	9.70	92	145	102.87	53
SO ₄	116	159.98	17.75	127	253	159.85	126
Alk	116	146.25	11.01	104.28	189.6	146.94	85.32

Table (4-5): Augmentation direct 2.

D2	COUNT	MEAN	STD	MIN	MAX	MEDIAN	RANGE
RC	348	2.33	1.73	0	6.06	1.8	6.06
Turb	348	6.66	4.18	0.87	33.88	5.69	33.01
PH	348	7.52	0.18	6.86	7.86	7.55	0.99
Temp	348	32.72	2.56	25	40.63	32.69	15.63
TDS	348	609.35	25.94	553	677.42	606.89	124.42
EC	348	920.57	50.23	837	1060.91	911.95	223.91
TH	348	344.72	31.53	276.72	430.35	340.28	153.63
Ca	348	79.19	13.54	44	112.12	77.45	68.12
Cl	348	105.49	9.69	91.67	145.76	102.69	54.08
SO ₄	348	159.93	17.77	126.57	253.22	160.20	126.65
Alk	348	146.21	10.96	103.17	189.6	146.94	86.42

The same pattern was observed for direct 2 in Table (4-6).

Table (4-6): t-test and P value result direct 2.

D2	T-test	P-value
RC	-0.027	0.97
Turb	0.025	0.97
PH	0.019	0.98
Temp	-0.0002	0.99
TDS	-0.021	0.98
EC	-0.005	0.99
TH	0.020	0.98
Ca	0.003	0.99
Cl	-0.026	0.97
SO ₄	0.022	0.98
Alk	0.027	0.97

4.4 The model outcomes

4.4.1 Random Forest Model

The RF regression model was built using a random subset data. The subset of data is randomly selected to construct trees, and ultimately, the average of the forecasts from all the trees gives the final prediction (**Biau & Scornet, 2016**). To assess the model's sensitivity to the variables, they were categorized into three groups with three scenarios: the first group includes dataset1, the second pertains to dataset 2, and the third relates to dataset 3. The three scenarios involved distance, time, and date. At the same time, the common hyperparameters for the three cases were n-estimator, maximum depth, minimum sample split, and minimum sample leaf. Where it was ascertained utilizing the GridSearchCV technique. The hyperparameters for the three scenarios are delineated in the Table (4-7) and Table (4-9) for two

directions. For direct 1 in the first scenario, all variables are present together and exhibit an R-square value of 0.919. Figure (4-14) illustrates the graph of actual values versus predicted values. In the second scenario, it represents the dataset 2 as illustrated in Figure (4-15), where performance decreased to 0.89. The third scenario represents the dataset 3 in Figure (4-16), where the R square has reached 0.92. Where the actual and predicted plot for the training sets of RF for direct 1 provided in Appendix B (Figure (B.1), Figure (B.2), Figure (B.3)).

As for direct 2 The superiority of the first scenario is clear, with an R square of 0.932 Figure (4.17), followed by the third scenario at R square 0.912 Figure (4.19) and the second scenario at R square 0.798 Figure (4.18). and for the training sets plot of RF for direct 2 provided in Appendix B (Figure (B.4), Figure (B.5), Figure (B.6)).

Concerning the errors as illustrated in Table (4-8) for direct 1, RMSE and MAE were utilized with MSE for enhanced explanation. The first scenario, including all variables, produced minimal errors in contrast with the other scenarios. Where the errors metrics indicate that this scenario is more accurate. In the second scenario, the error metrics indicating a significant rise in error magnitude. And the third scenario enhances the amount of error in the second scenario, indicating nearly identical performance to the first scenario. For the direct 2 illustrated in Table (4-10), the first scenario, the errors metrics indicate that this scenario is more accurate.

In the second scenario, the error metrics indicating a significant rise in error magnitude. The third scenario enhances the amount of error in the second scenario, indicating performance is nearly equivalent to the first scenario.

Based on the performance of the models above, it shows that eliminating the features by removing dataset 3 in the second scenario will reduce the model's efficacy and result in diminished performance. Nonetheless, after employing all the variables, the performance difference was relatively negligible compared to the third scenario. Additionally, in direct 2, the third scenario has high performance, but still, the scenario one with all variables is better. This suggests that the dataset 2 exert minimal influence in this model. In general, the model demonstrated excellent efficiency, with the exception of the second scenario, in which its performance was deemed acceptable and needed modification. In a study by **Onyutha (2022)** aimed at predicting RC, the RF model had superior performance relative to the other models employed, achieving an R^2 of 0.66 for the test data. And a study by **Kyritsakas et al. (2023)** predicted chlorine loss events just within the primary pipeline network for DW in three different locations in the UK. Although the target variables differ between the current study and this research, the comparison remains valid by evaluating the model's performance. The performance metrics indicated a range for the three pipes: MAE (0.0205-0.0385) and RMSE (0.02645-0.048). These findings demonstrate high accuracy for the RF model, indicating its reliability in predicting chlorine tasks across the network for different situations.

Table (4-7): the hyperparameters for RF direct 1.

RF_D1	Scenario 1	Scenario 2	Scenario 3
n-estimator	200	200	100
Max. depth	20	20	30
Min. sample split	2	2	2
Min. sample leaf	1	1	1

In general, the RF in the current study provided excellent predictive ability for RC in DW, explaining more than 90% of the variation in some scenarios. The current study differs from previous ones due to data quality and processing. Therefore, findings suggest the model's potential for enhancement when appropriate data is available.

This study's conclusions significantly improve the confidence of the RF model in the prediction of RC.

Table (4-8): Performance metric for RF direct 1.

RF_D1	Scenario 1	Scenario 2	Scenario 3
R ²	0.919	0.897	0.922
MSE	0.205	0.263	0.197
MAE	0.333	0.371	0.343
RMSE	0.453	0.513	0.444

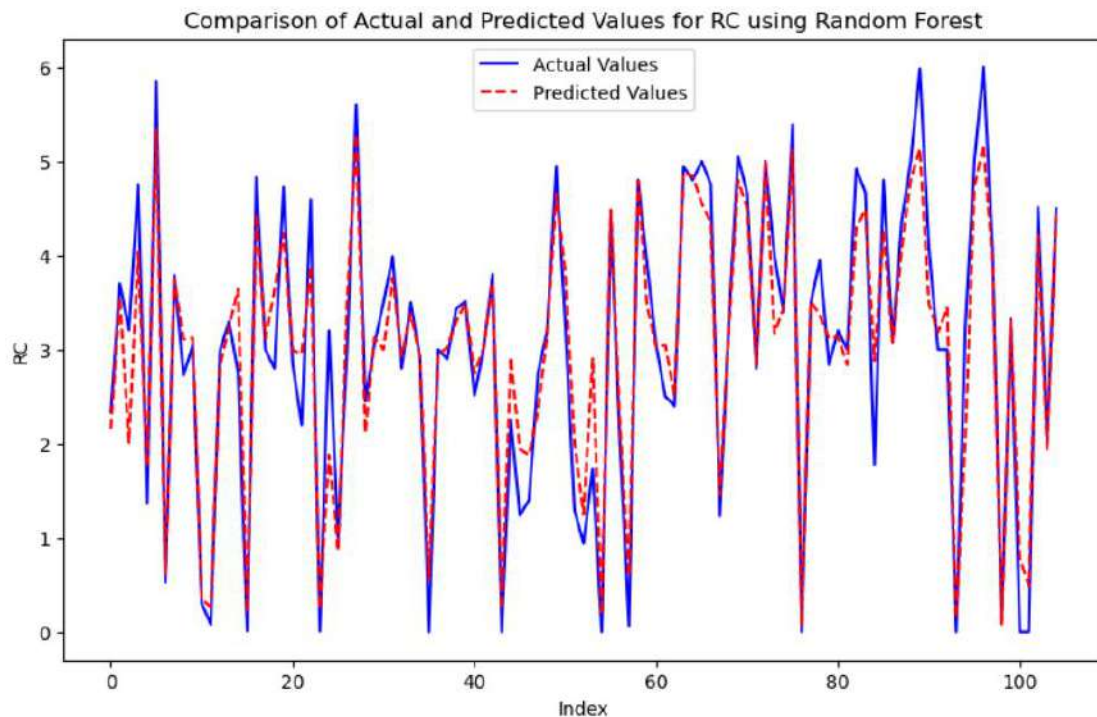


Figure (4-14): Actual and Predicted Values for RC using RF S1 direct1.

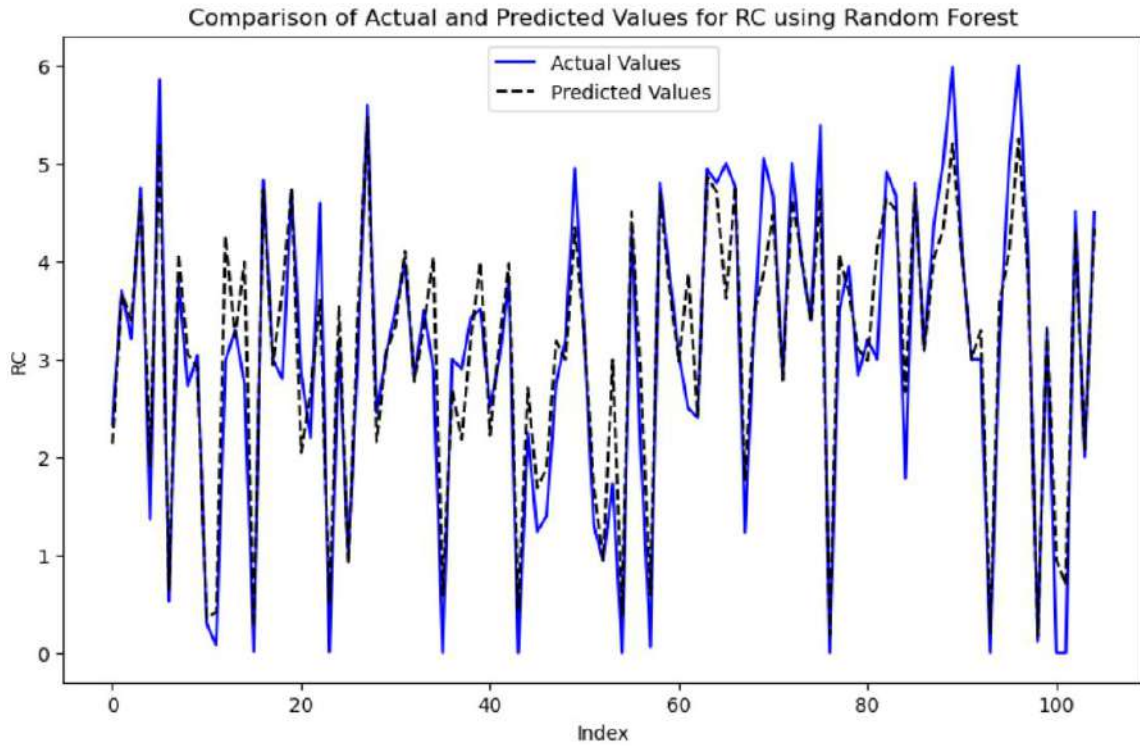


Figure (4-15): Actual and Forecasted Values for RC using RF S2 direct1.

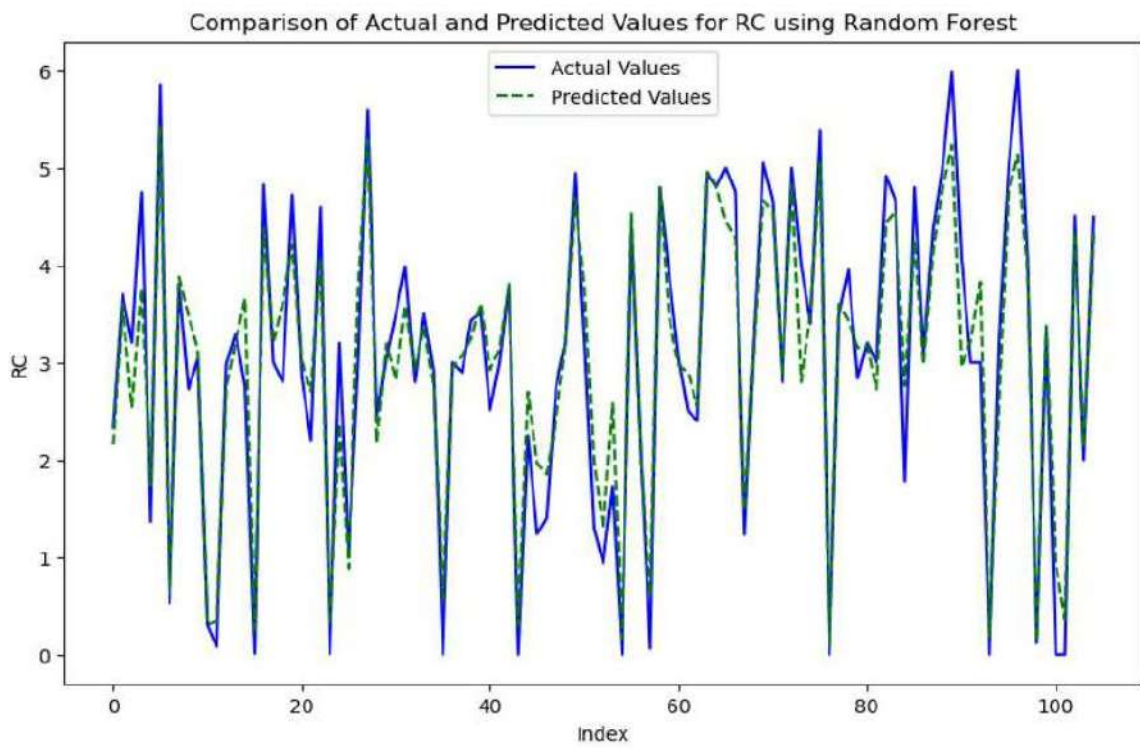


Figure (4-16): Actual and Predicted Values for RC using RF S3 direct1.

Table (4-9): the hyperparameters for RF direct 2

RF_D2	Scenario 1	Scenario 2	Scenario 3
n-estimator	200	200	200
Max. depth	30	30	30
Min. sample split	2	2	2
Min. sample leaf	1	1	1

Table (4-10): Performance metric for RF direct 2

RF_D2	Scenario 1	Scenario 2	Scenario 3
R ²	0.932	0.798	0.912
MSE	0.199	0.595	0.259
MAE	0.283	0.463	0.351
RMSE	0.446	0.771	0.508

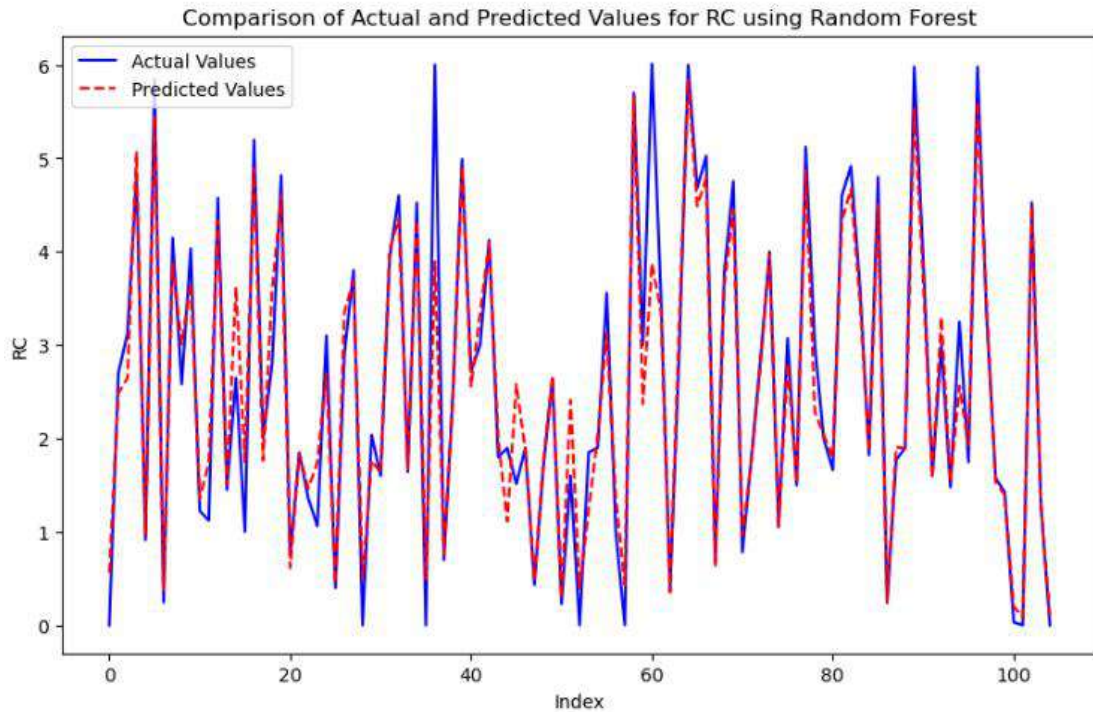


Figure (4-17): Actual and Predicted Values for RC direct 2_ S1

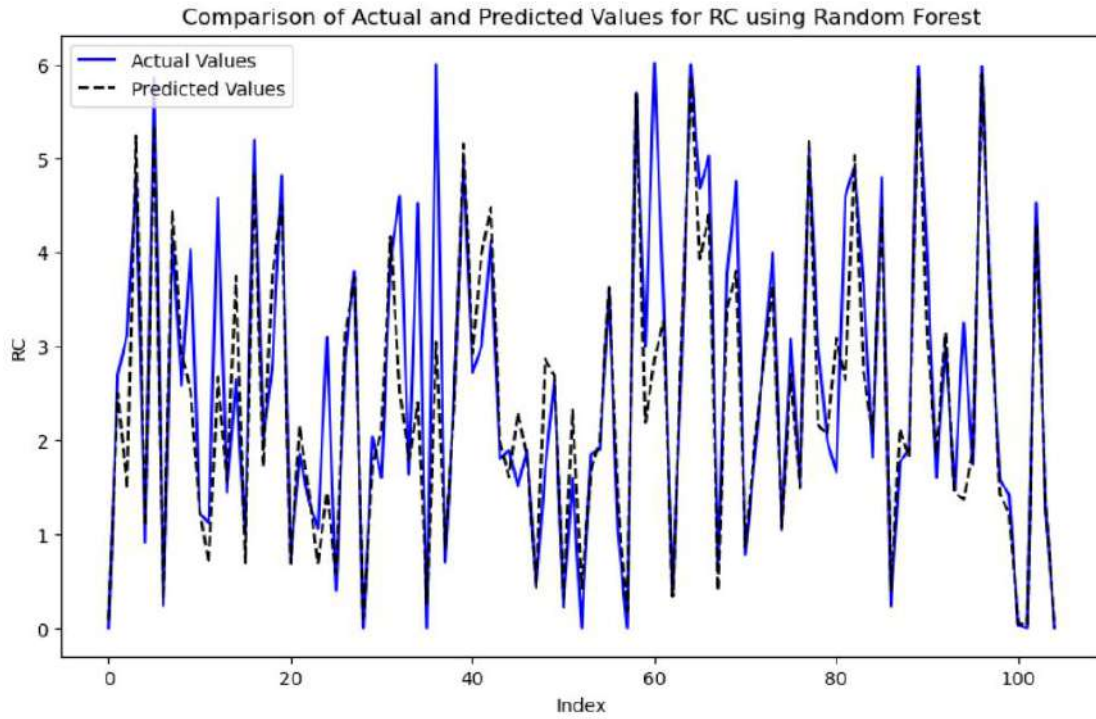


Figure (4-18): Actual and Forecasted Values for RC direct 2_S2

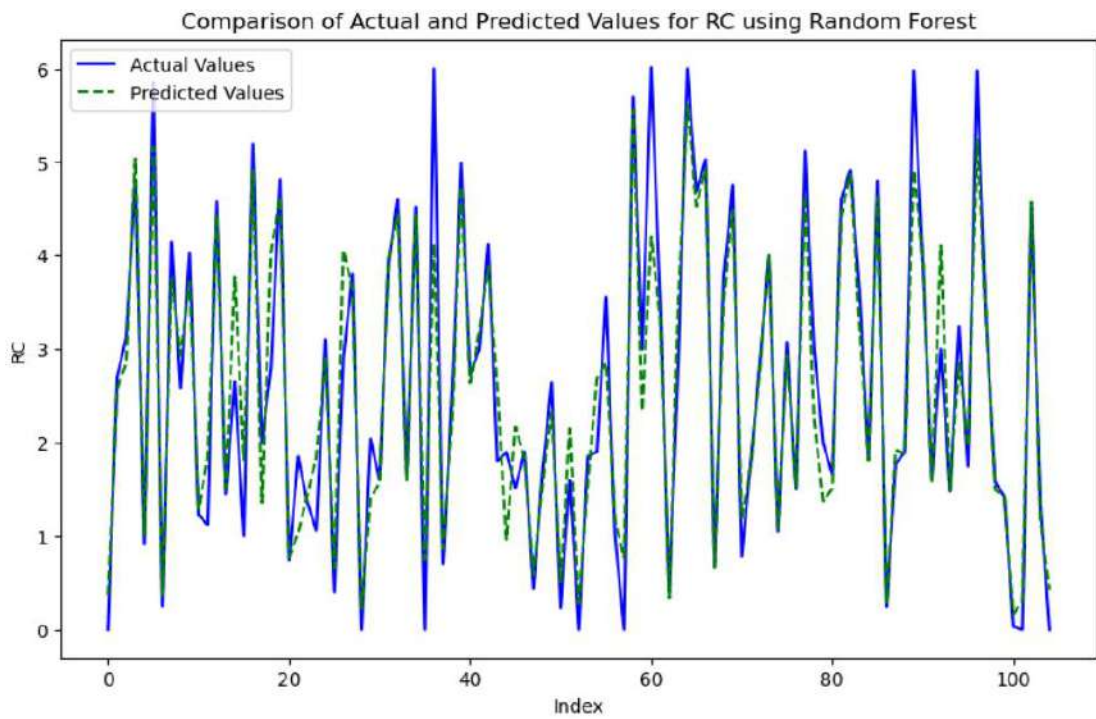


Figure (4-19): Actual and Predicted Values for RC direct 2_S3

4.4.2 XGBoost Regression model

The Extreme Gradient Boosting model has been developed and is considered one of the ML models. It is based on decision trees (**Ali et al., 2023**). The hyperparameters utilized to generate suitable combinations for the model in GridSearchCV are n estimators, maximum depth, learning rate, subsample, and colsample by tree. The hyperparameters for all scenarios are delineated in the Table (4-11) and Table (4-13) for two directions.

For direct 1 in the first scenario, display an R-square value of 0.959 Figure (4-20) illustrates the graph of actual values versus predicted values. In the second scenario, illustrated in Figure (4-21), where performance decreased to 0.937. The third scenario in Figure 4-22, where the R square has reached 0.929. (See Appendix B, Figure B.7, B.8 and B.9).

Concerning direct 2, The first scenario demonstrates an R-squared value of 0.965 as shown in Figure (4-23). Depict the actual values with the predicted values, followed by the third scenario at 0.964, illustrated in the Figure (4-24). And the second scenario at 0.886 Figure (4-25). Significantly, the first and third scenarios approximately demonstrate the same performance with minimal variation. The second scenario had the least performance. And plot training plot (See Appendix B, Figure B.10, B.11 and B.12).

Regarding the errors as illustrated in Table (4-12), for the direct 1 the first scenario, the errors metrics demonstrate that this scenario seems more accurate due to less error. In the second scenario (dataset 2), the error magnitude has slightly increased. However, the third scenario (dataset 3) enhances the amount of error in the second scenario.

In addition, for direct 2 illustrated in Table (4-14), the first scenario indicate that this scenario is more accurate. In the second scenario, the error metrics

indicating a significant rise in error magnitude. And the third scenario enhances the amount of error in the second scenario, demonstrating performance is nearly equivalent to the first scenario.

Based on the performance of the models above, it shows that since the XGBoost model utilizes gradient boosting, it exhibits greater sensitivity to fine data details in contrast to other models. XGBoost showed high predictive accuracy in both directions with the first scenario. Scenarios two and three reflect varying perspectives about which is best. The direct one demonstrates the superiority of the dataset 2 model relative to the dataset 3 model. However, the second direction indicated that the dataset 3 model surpassed the dataset 2 model. This is because the data in the two pipes for the two directions differ in terms of distances and times, which causes them to vary, in addition to the different field measurement conditions between them, resulting in this difference in the data. In general, the XGBoost model has consistently demonstrated accuracy in predicting the residual free chlorine in every scenario. Because there was no similar study to predict RC using XGBoost for comparison purposes, it has been used to predict DW quality in general in previous literature. According to a study conducted by **(Joslyn, 2018)**, the model showed performance ranging between 81.64% and 98.49%.

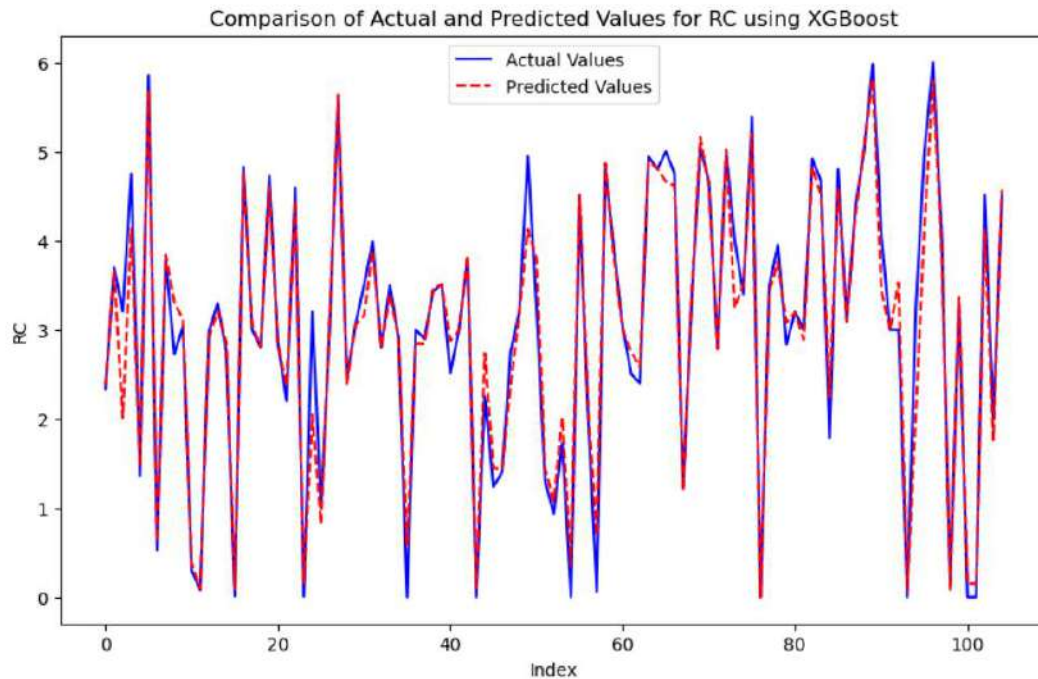
Table (4-11): the hyperparameters for XGBoost direct 1.

XGBoost_D1	Scenario 1	Scenario 2	Scenario 3
n-estimator	200	200	200
Max. depth	9	6	6
Learning rate	0.1	0.1	0.2
subsample	0.8	0.8	0.8
Colsample by tree	0.8	1	0.8

Table (4-12): Performance metric for XGBoost direct 1.

XGBoost_D1	scenario1	Scenario2	Scenario 3
R^2	0.959	0.937	0.929
MSE	0.103	0.161	0.180
MAE	0.199	0.265	0.278
RMSE	0.321	0.401	0.424

Where its performance is regarded as superior. And due to the model's speed, it is seen as an essential tool for tasks needing high performance and fast warnings. This improves the model's reliability and enables its continued use in investigations on the prediction of RC at critical locations.

**Figure (4-20):** Actual and Forecasted Values for RC using XGBoost S1direct1

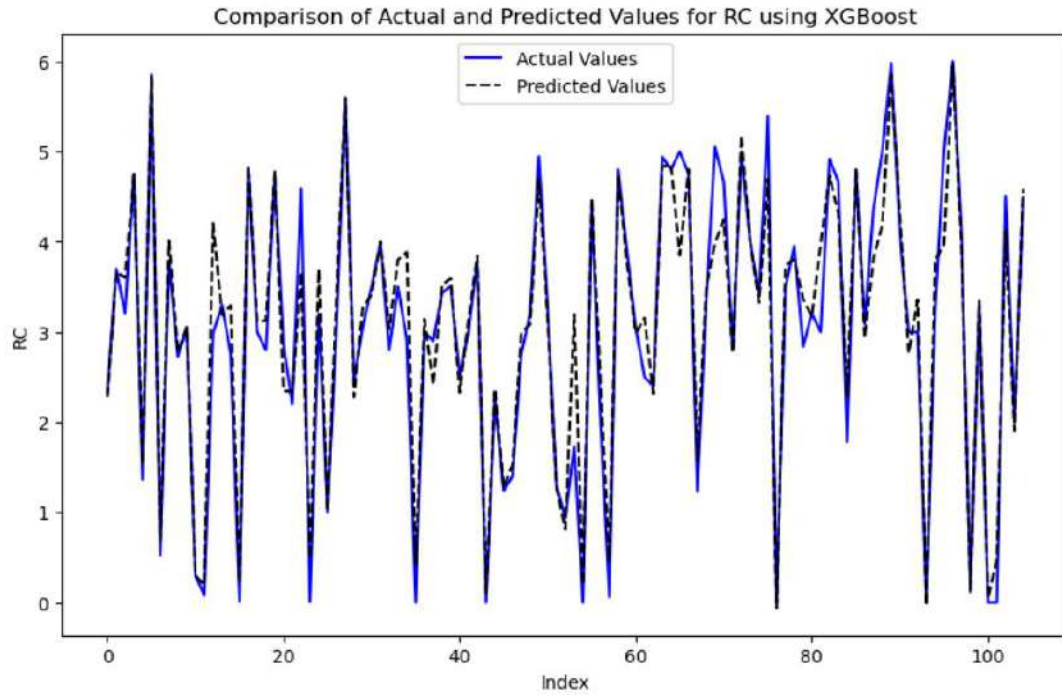


Figure (4-21):Actual and Forecasted Values for RC using XGBoost S2 direct1.

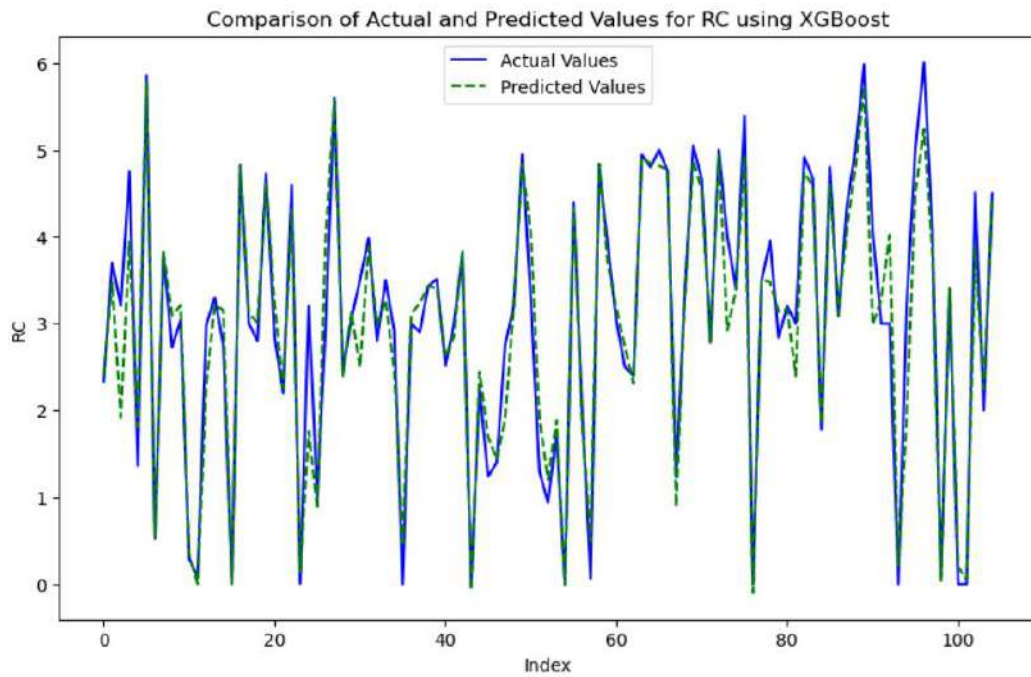


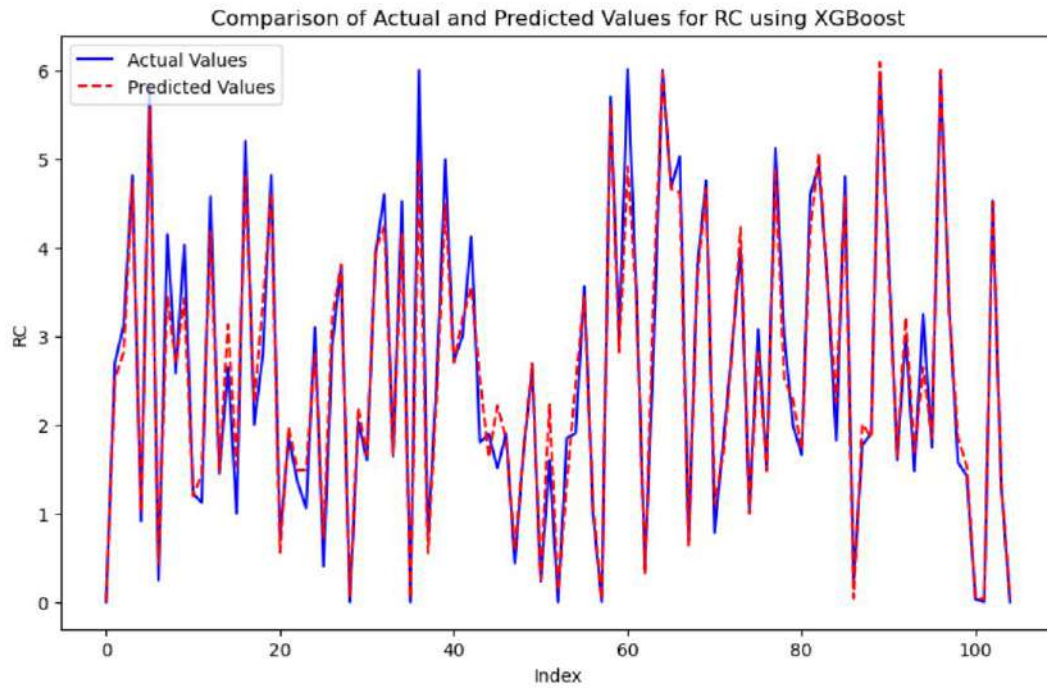
Figure (4-22): Actual and predicted values for RC by XGBoost S3 direct1.

Table (4-13): the hyperparameters for XGBoost direct 2.

XGBoost_D2	Scenario 1	Scenario 2	Scenario 3
n-estimator	200	200	200
Max. depth	3	9	6
Learning rate	0.1	0.1	0.1
subsample	1	0.8	0.8
Colsample by tree	0.1	1	0.8

Table (4-14): Performance metric for XGBoost direct 2.

XGBoost_D2	scenario1	Scenario2	Scenario 3
R ²	0.965	0.886	0.964
MSE	0.102	0.335	0.105
MAE	0.229	0.314	0.214
RMSE	0.319	0.579	0.324

**Figure (4-23):** Actual and Forecasted Values for XGBoost direct 2_S1.

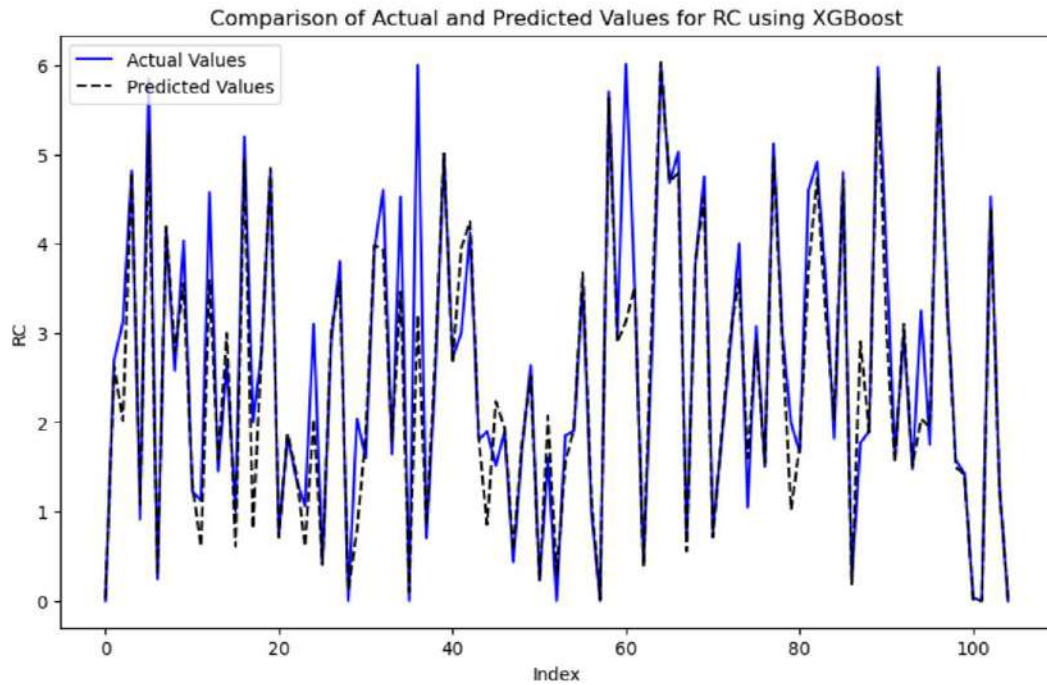


Figure (4-24): Actual and Forecasted Values for XGBoost direct 2_S2.

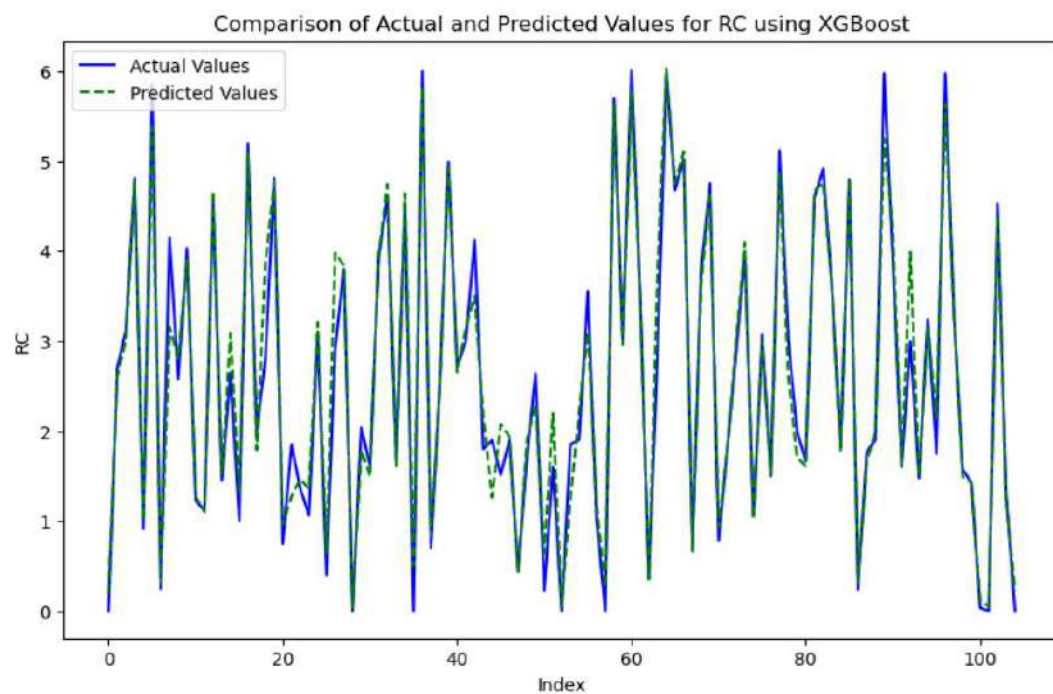


Figure (4-25): Actual and Forecasted Values for XGBoost direct 2_S3.

4.4.4 MLP Regression

The MLP is a form of ANN that represents a type of deep learning due to its inclusion of numerous hidden layers in this model (Gupta, 2018). The principle of three scenarios, as previously expressed, has been applied. Following this model hyperparameters were controlled using GridSearchCV, which included the activation function, alpha, and hidden layer sizes and numbers. Table (4-15) and Table (4-17) show the hyperparameters for the first and second directions within the model. For the first direction, as shown in Figure (4-27), the first scenario demonstrated all features entering the network and has excellent results with an R-squared of 0.917. Followed by the third scenario that includes dataset 3 as inputs Figure (4-31), which has an R-squared of 0.847, while the second scenario input of dataset 2 in Figure (4-29) exhibits the lowest efficacy with an R-squared of 0.7878. (See Appendix B, Figure B.13, B.14 and B.15).

In the case of the second direction, the third scenario in Figure (4-36). demonstrated superiority with an R-squared of 0.96, while the first scenario in Figure (4-33). exhibited an R-squared of 0.955. It is clear that the disparity between the third and first instances is slight; their performance is close to identical. The second scenario in Figure (4-35). presents an R-squared of 0.80, demonstrating that it is the lowest effective performance. (See Appendix B, Figure B.16, B.17 and B.18).

Regarding the errors as illustrated in Table (4-16) for the direct 1, the first scenario produced minimal errors in contrast with the other scenarios. In the second scenario, the magnitude of errors is increasing. It is clear that the error numbers have increased, which is somewhat undesirable. While the third scenario, the error values return to be lower than in the previous scenario.

As for the direct 2 illustrated in Table (4-18), the first scenario appears to have the smallest errors, close to zero. In the second scenario, the number of errors also increased. Meanwhile, the third scenario enhances the number of errors in the previous scenario.

All in all, the model experienced a strong capacity to predict RC in DW, with exceptional performance in the first direction, with the inclusion of all features. Figure (4-26). demonstrates the model's ability to correlate actual values with predicted values. And in the second scenario, the performance reduced with the removal of the dataset 3, resulting in a value of 0.78, which means a failure to explain about 22% of the variance in the data, as illustrated in Figure (4-28), where the disparity that occurred between the actual and predicted results increases. When evaluating the model using the dataset 3, its performance was enhanced to 84%, as represented in Figure (4-30), which clearly demonstrates the performance. Consequently, the dataset 3 explain an important part of the enhancement in the accuracy of the model. In the second direction, the performance of the first scenario, illustrated in Figure (4-32), was excellent; however, the third scenario slightly surpassed the first one, as shown in Figure (4-36). This is explained by the difference in both directions in terms of noise in the data. As for the second scenario, illustrated in Figure (4-34), both directions agree that it exhibits the lowest performance among the scenarios. This signifies that dataset 2 alone are inadequate. In general, the MLP model exhibited excellent results in both the first scenario and the third scenario from the second direction, aligning according to a recent study **(Batista et al., 2024)** that focused on creating a multi-layered network to assess the deterioration of free RC in Brazilian water distribution systems, aiming to forecast failures or variances from requirements and to simplify

early adjustments to chlorine dosages. It achieved an accuracy rate of up to 99.84%.

The model exhibited its ability to fix the unreliability of the kinetic equation for the reaction between chlorine and contaminants. It also eliminates the necessity for laboratory experiments to determine the chlorine degradation coefficient.

Table (4-15): the hyperparameters for MLP direct 1.

MLP_D1	Scenario 1	Scenario 2	Scenario 3
Activation Function	relu	relu	relu
alpha	0.001	0.0001	0.0001
Hidden layer sizes	(128,64,32)	(128,64,32)	(128,64,32)
Solver	Adam	Adam	Adam

Table (4-16): Performance metric of MLP direct1.

MLP_D1	scenario1	Scenario2	Scenario 3
R ²	0.917	0.787	0.847
MSE	0.242	0.625	0.45
MAE	0.299	0.362	0.343
RMSE	0.492	0.791	0.670

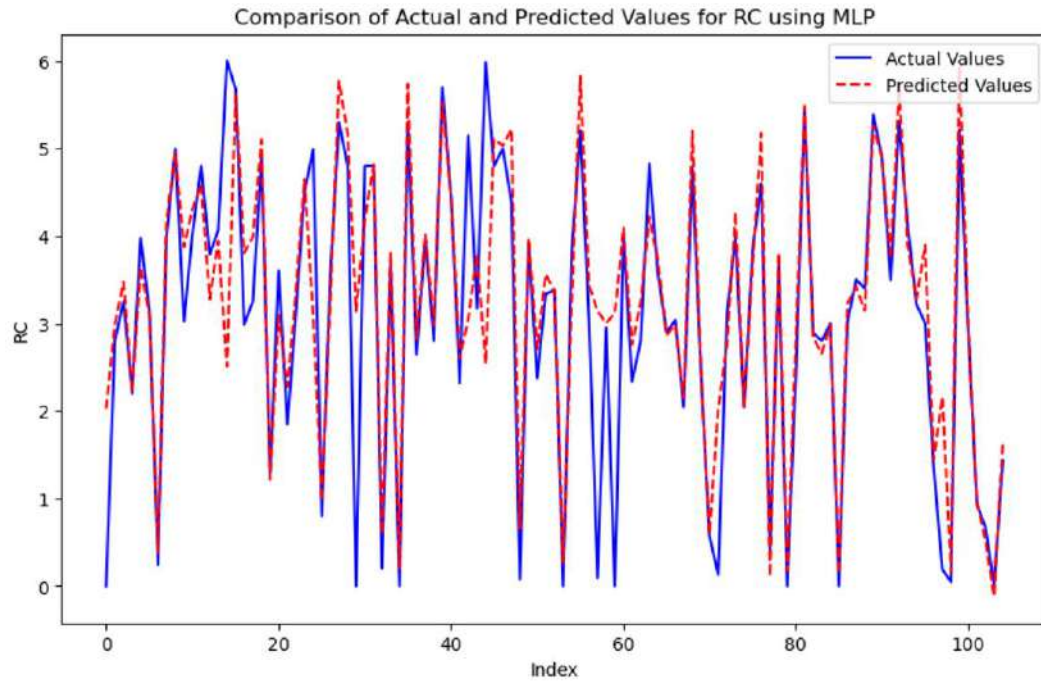


Figure (4-26): Actual and Forecasted Values for RC by MLP for S1 direct1.

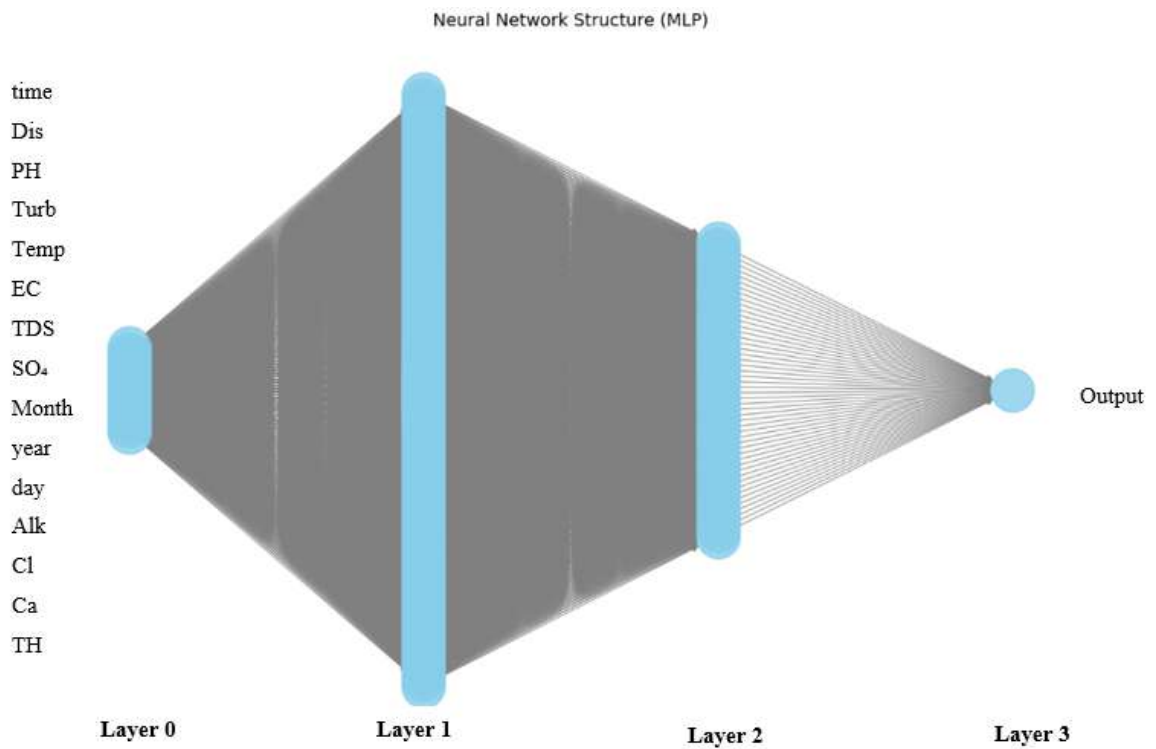


Figure (4-27): Neural Network Structure (MLP) for S1direct1.

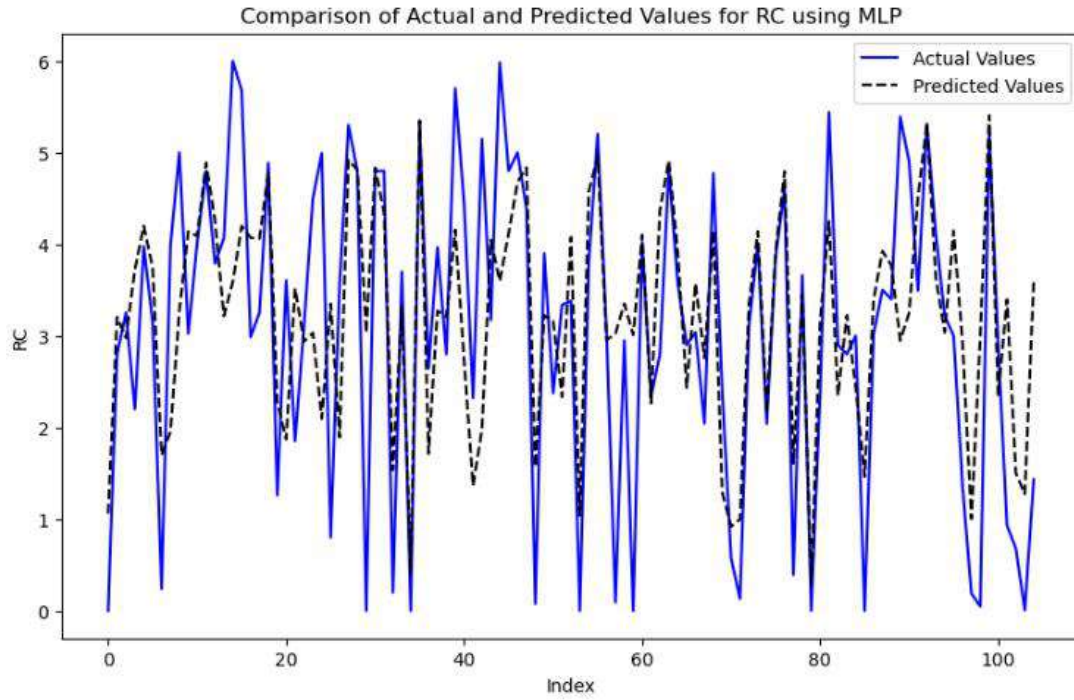


Figure (4-28): Actual and Forecasted Values for RC by MLP for S2 direct1.

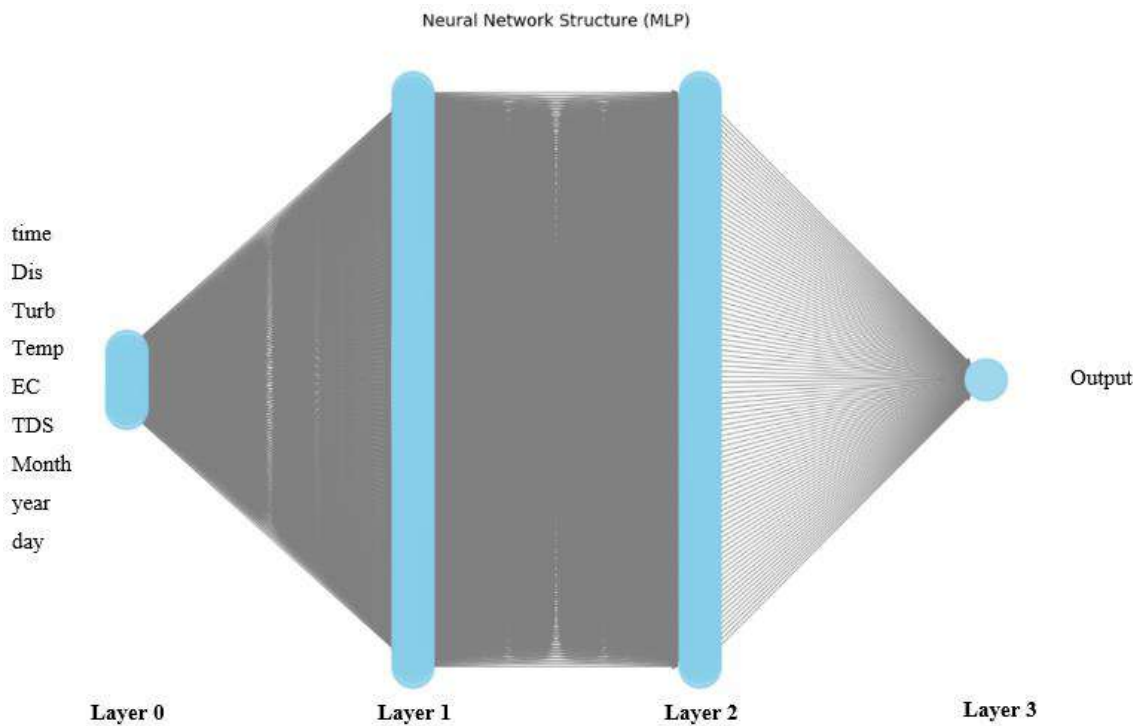


Figure (4-29): Neural Network Structure (MLP) for S2 direct1.

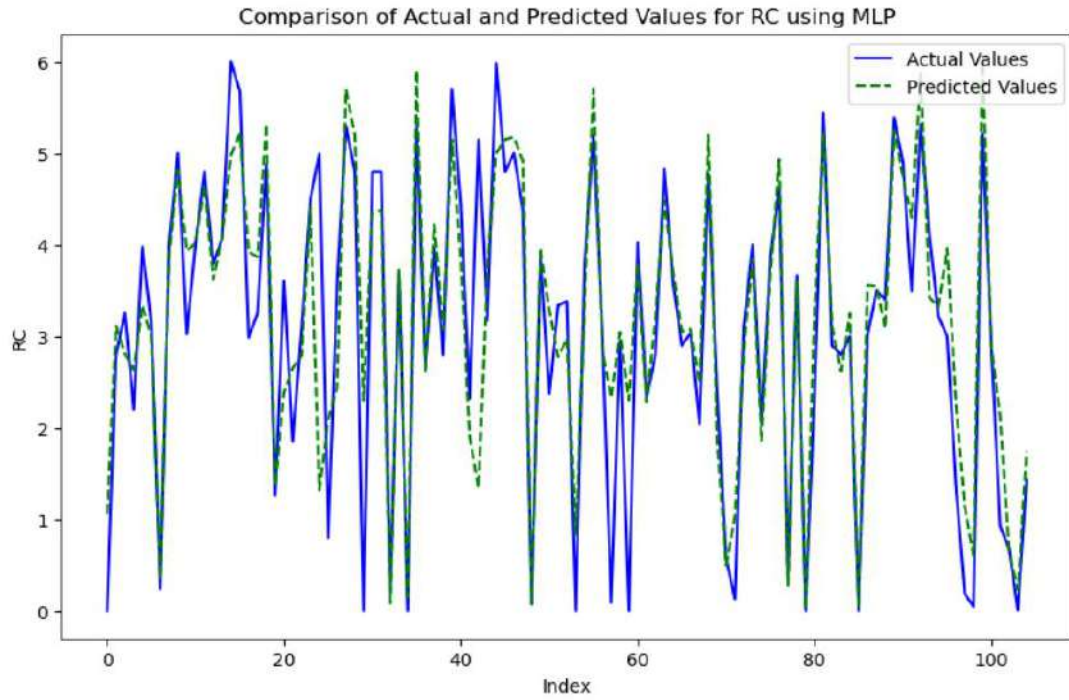


Figure (4-30): Actual and Forecasted Values for RC by MLP for S3 direct1.

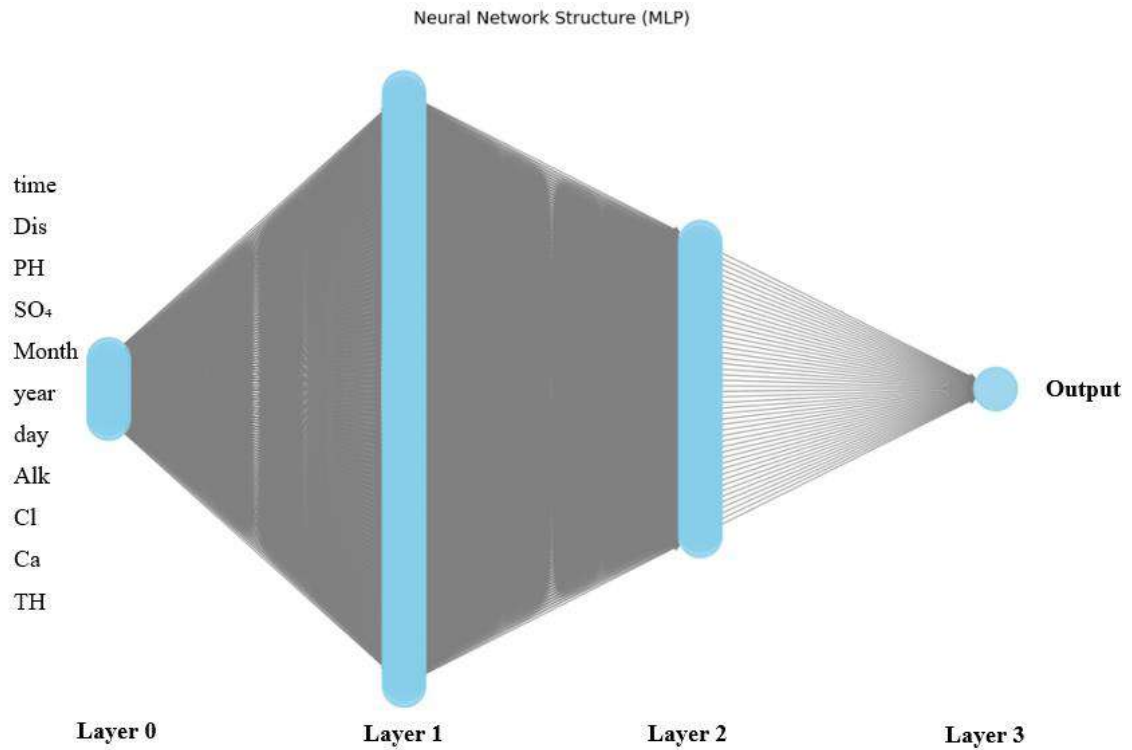


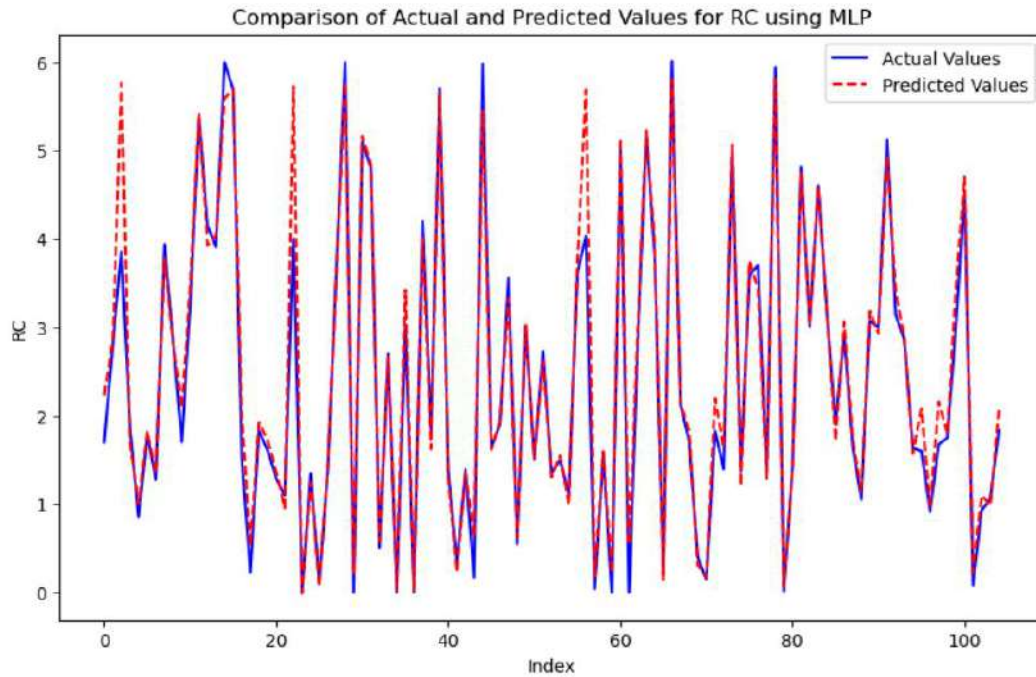
Figure (4-31): Neural Network Structure (MLP) for S3 direct1.

Table (4-17): the hyperparameters for MLP direct 2.

MLP_D2	Scenario 1	Scenario 2	Scenario 3
Activation Function	relu	relu	relu
alpha	0.001	0.0001	0.0001
Hidden layer sizes	(128,64,32)	(128,64,32)	(128,64,32)
Solver	Adam	Adam	Adam

Table (4-18): Performance metric of MLP direct 2

MLP_D2	scenario1	Scenario2	Scenario 3
R ²	0.955	0.803	0.960
MSE	0.132	0.586	0.117
MAE	0.208	0.552	0.194
RMSE	0.363	0.765	0.342

**Figure (4-32):** Comparison of Actual and Forecasted Values for RC by MLP for S1 direct2.

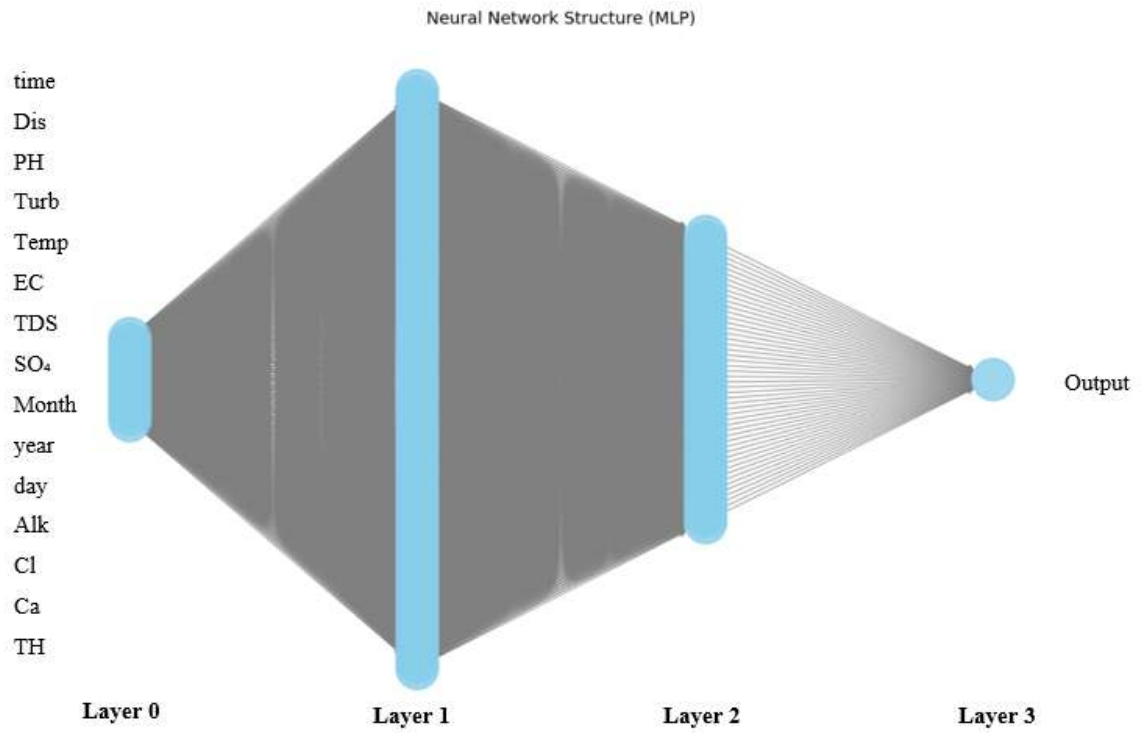


Figure (4-33): Neural Network Structure (MLP) for S1 direct2.

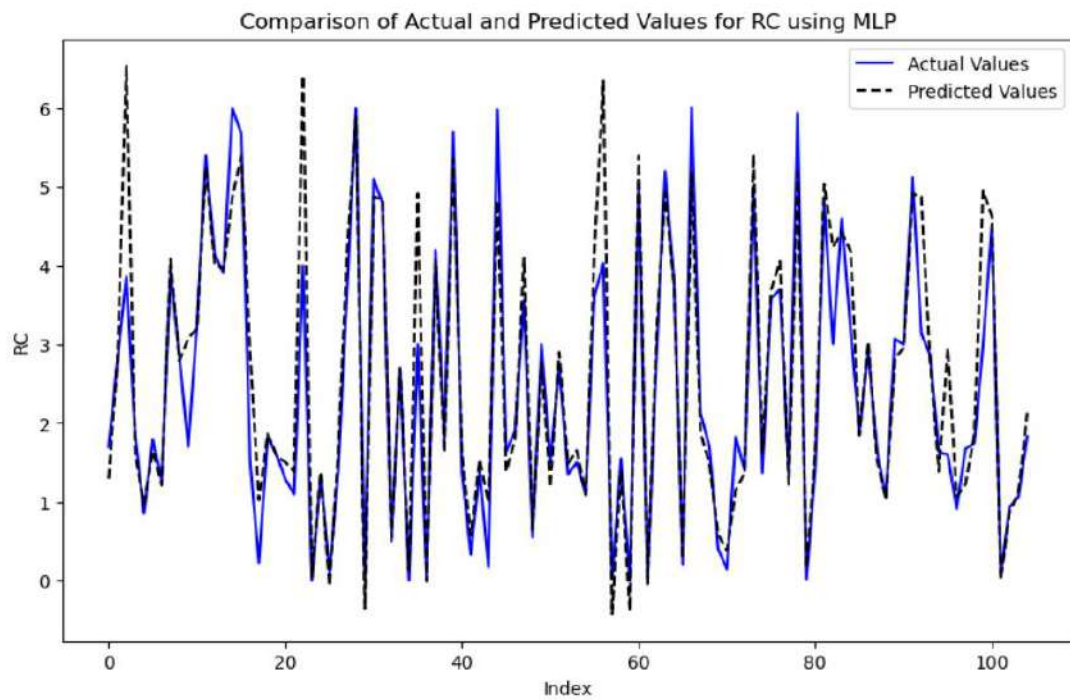


Figure (4-34): Comparison of Actual and Forecasted Values for RC by MLP for S2 direct2.

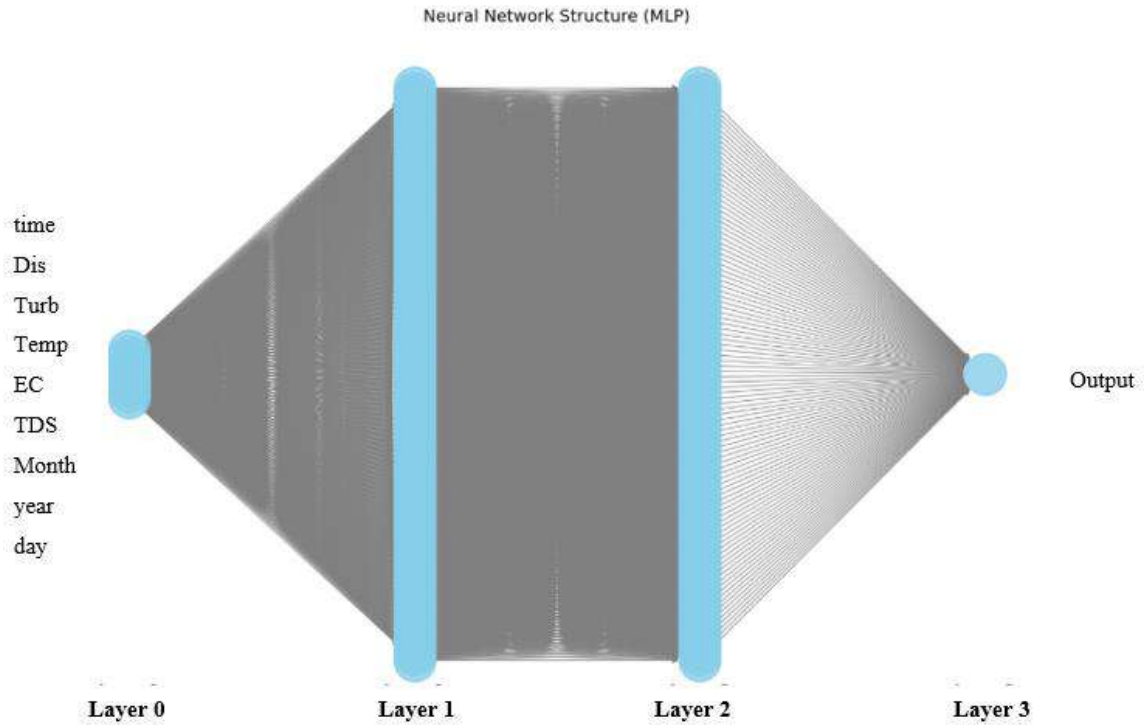


Figure (4-35): Neural Network Structure (MLP) for S2 direct2.

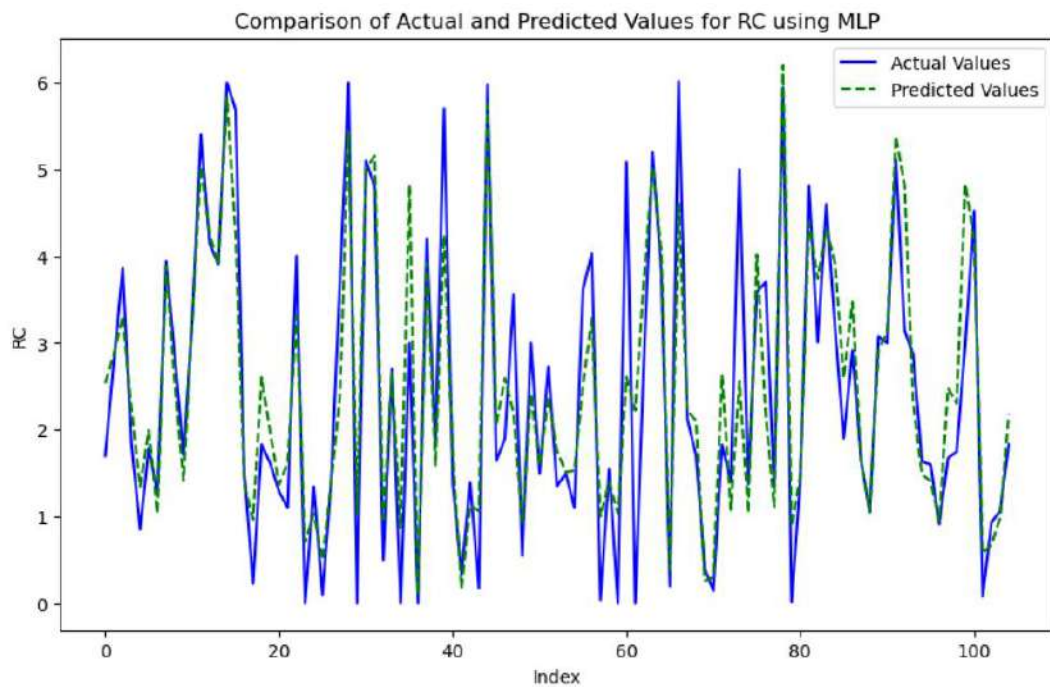


Figure (4-36): Comparison of Actual and Forecasted Values for RC by MLP for S3 direct2.

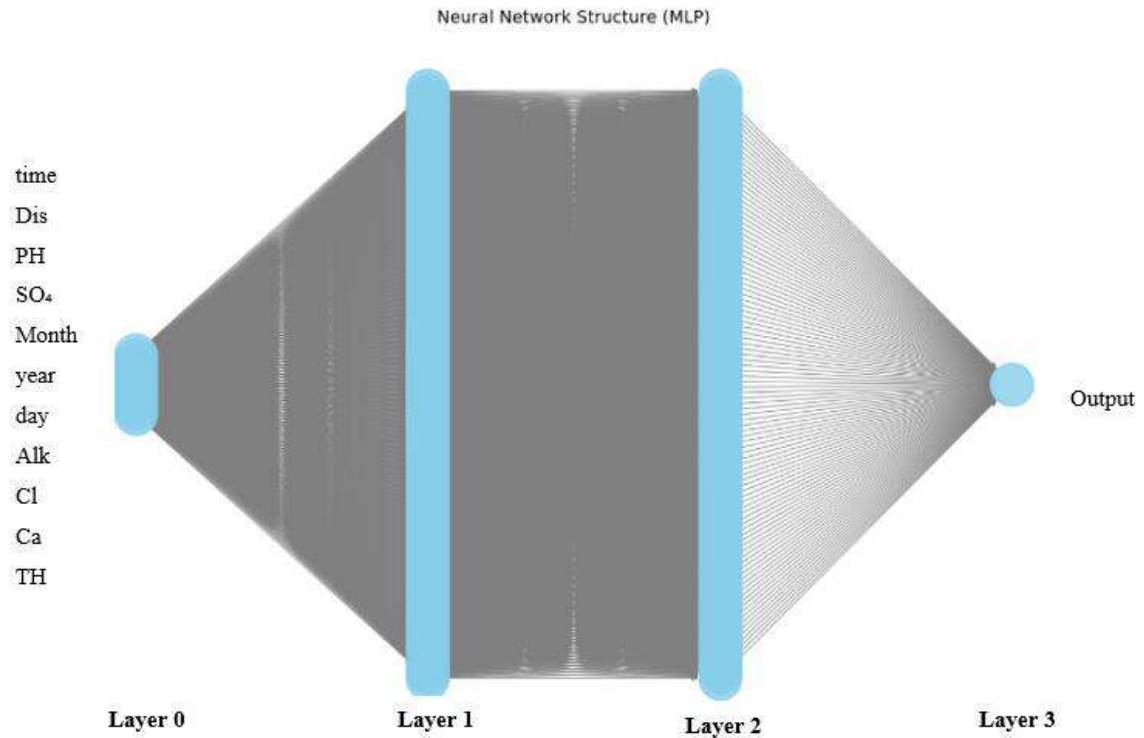


Figure (4-37): Neural Network Structure (MLP) for S3 direct2.

4.4.5 GRNN Model

A feedforward ANN model, which is a GRNN, was constructed, utilizing the principle of non-linear regression in its function. The design of GRNN comprises four layers: input, pattern, summation, and output. As for the sigma value, it was modified within the network, considering it the only one hyperparameter subject to modification in the code, through a process of trial and error.

Regarding the performance metrics for direct 1 presented in Table (4-20), where the neural network's findings are illustrated in a manner similar to the other models categorized after running the model for general regression across three scenarios. The outcomes of the first scenario are depicted in Figure 4-38, illustrating both the actual and predicted numbers. The model

showed an R-squared of 0.996, and for the second scenario, it exhibited an R-square value of 0.948 Figure (4-40). In the third scenario, the R-squared value increased to 0.995 Figure (4-42). Furthermore, Figure (4-39), Figure (4-41) and Figure (4-43) represent the structure of the GRNN network for the three scenarios in the first direction, respectively, illustrating the variations in inputs among various scenarios. (See Appendix B, Figure B.19, B.20 and B.21).

Regarding direct 2, the first scenario exhibits supremacy with an R-square of 0.991, as shown in Figure (4-44), which displays the actual along with the predicted values. Next follows the third scenario, which has an R-squared of 0.947, illustrated in Figure (4-48). and the second scenario gives R-square equal to 0.925, Figure (4-46). In addition, Figure (4-45), Figure (4-47) and Figure (4-49) show the structures of the GRNN network for the three scenarios in the second direction, respectively. (See Appendix B, Figure B.22, B.23 and B.24).

Table (4-19): the hyperparameters for GRNN direct 1.

GRNN_D1	Scenario 1	Scenario 2	Scenario 3
σ	0.2	0.2	0.2

Table (4-20): Performance metric of GRNN direct 1.

GRNN_D1	Scenario 1	Scenario 2	Scenario 3
R ²	0.996	0.948	0.995
MSE	0.009	0.132	0.010
MAE	0.068	0.124	0.069
RMSE	0.098	0.364	0.102

Considering the errors shown in Table (4-19) for the direct 1, The first scenario, indicates a higher level of accuracy. And in the second

scenario, the magnitude of the inaccuracy has slightly increased. While the third scenario improves the number of errors present in the second scenario. As for the direct 2 shown in Table (4-22), the first scenario metrics suggest that this scenario exhibits more accuracy. And in the second scenario, signifying a substantial increase in error magnitude. The third scenario increases the error in the second scenario.

Based on the performance of the models above, it shows that the first scenario demonstrated superiority, succeeded by the third scenario, and followed by the second scenario in both directions. The model exhibited exceptional accuracy across every scenario, even with variations within cases, as the performance almost aligns with the study by **Wadkar & Kote (2017)** in India, where the GRNN model proved superior efficacy in forecasting RC levels in a plant for water treatment, achieving excellent results in the following metrics: $R\text{-Square}=0.979$, $MSE=0.001$, $MAE=0.019$. The model's success in achieving predicted values to actual values proves its ability to explain the data variation, indicating its efficacy in predicting RC in DW.

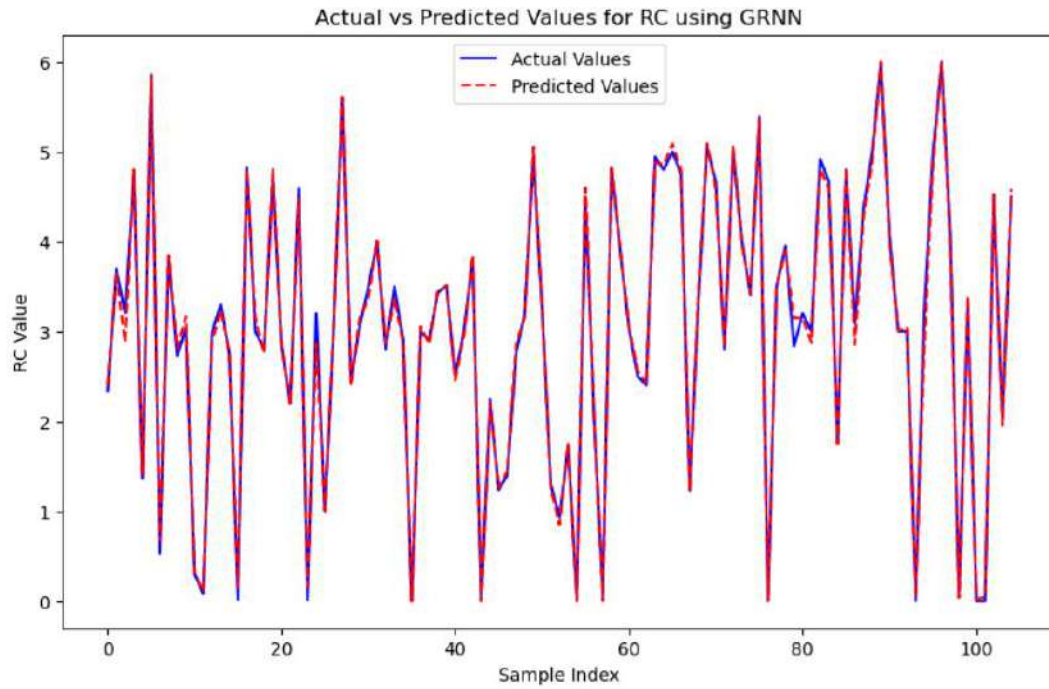


Figure (4-38): Actual vs Predicted Values for RC using GRNN for S1 direc1.

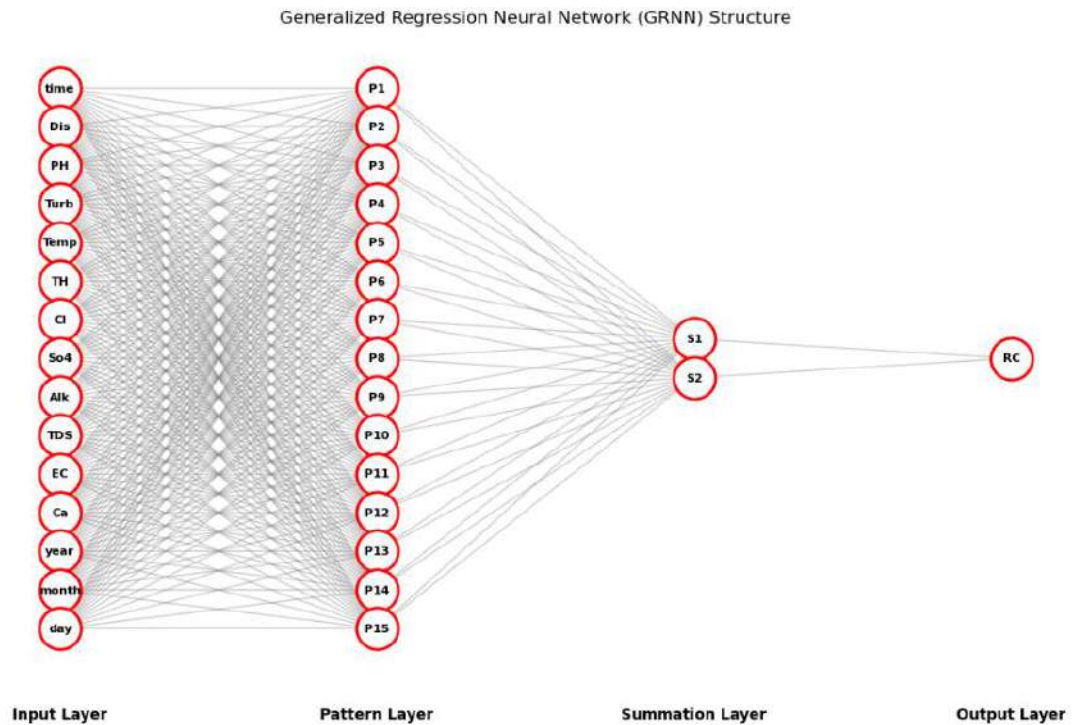


Figure (4-39): Generalized Regression Neural Network (GRNN) Structure for S1 direc1.

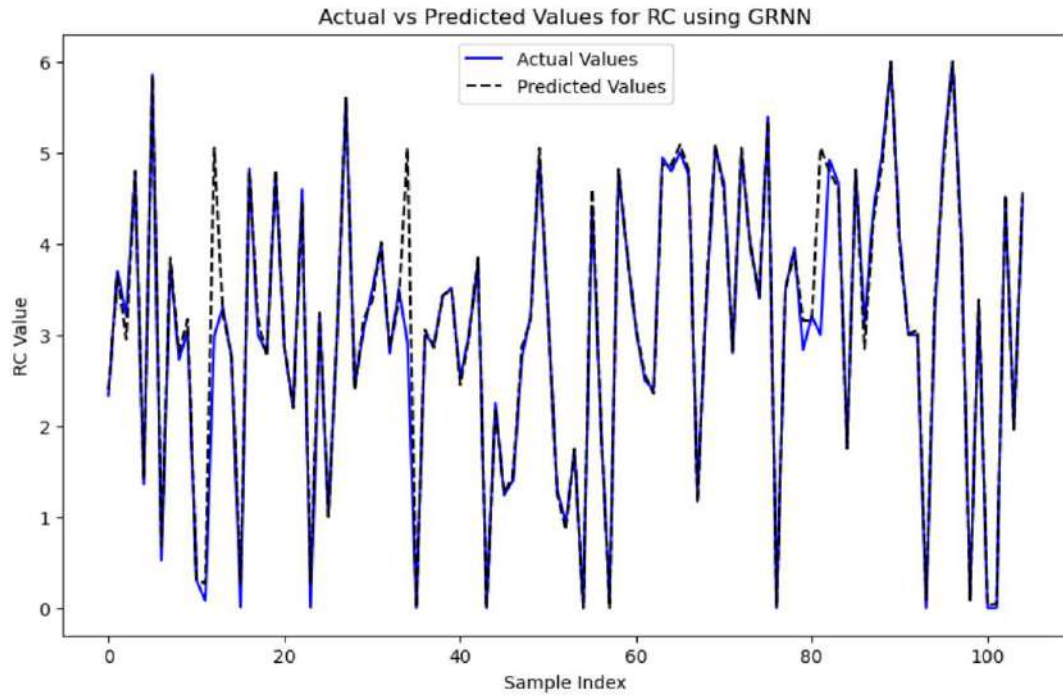


Figure (4-40): Actual vs Predicted Values for RC using GRNN for S2 direc1.

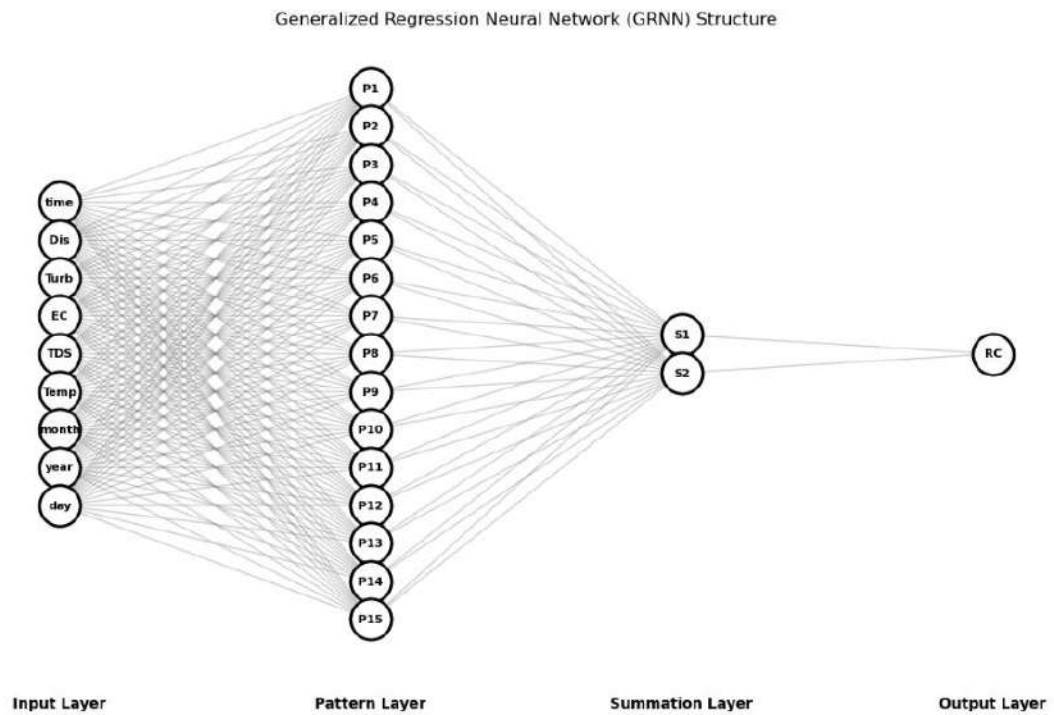


Figure (4-41): Generalized Regression Neural Network (GRNN) Structure for S2 direc1.

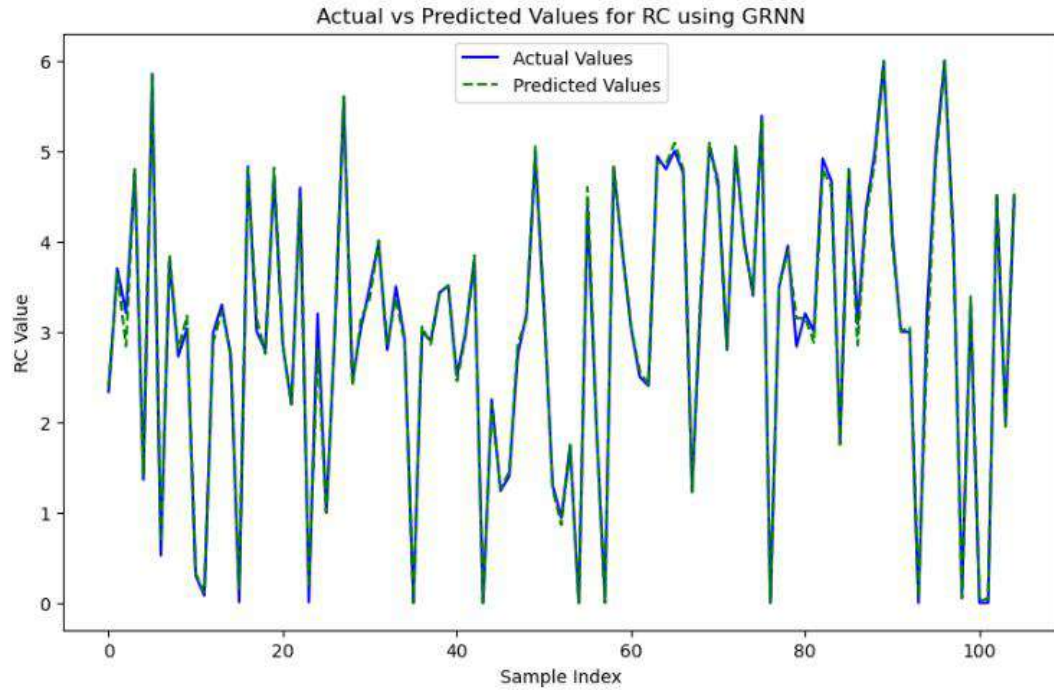


Figure (4-42): Actual vs Predicted Values for RC using GRNN for S3 direc1.

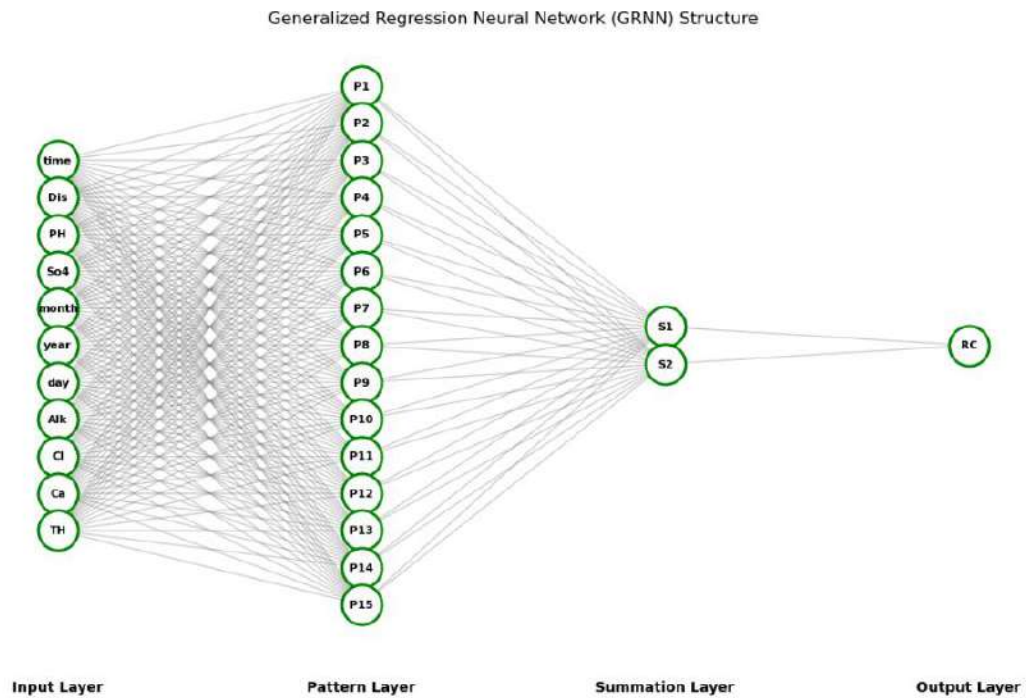


Figure (4-43): Generalized Regression Neural Network (GRNN) Structure for S3 direc1.

Table (4-21): the hyperparameters for GRNN direct 2.

GRNN_D2	Scenario 1	Scenario 2	Scenario 3
σ	0.2	0.2	0.2

Table (4-22): Performance metric for GRNN direct 2.

GRNN_D2	Scenario 1	Scenario 2	Scenario 3
R^2	0.991	0.925	0.947
MSE	0.024	0.219	0.156
MAE	0.099	0.171	0.152
RMSE	0.156	0.468	0.395

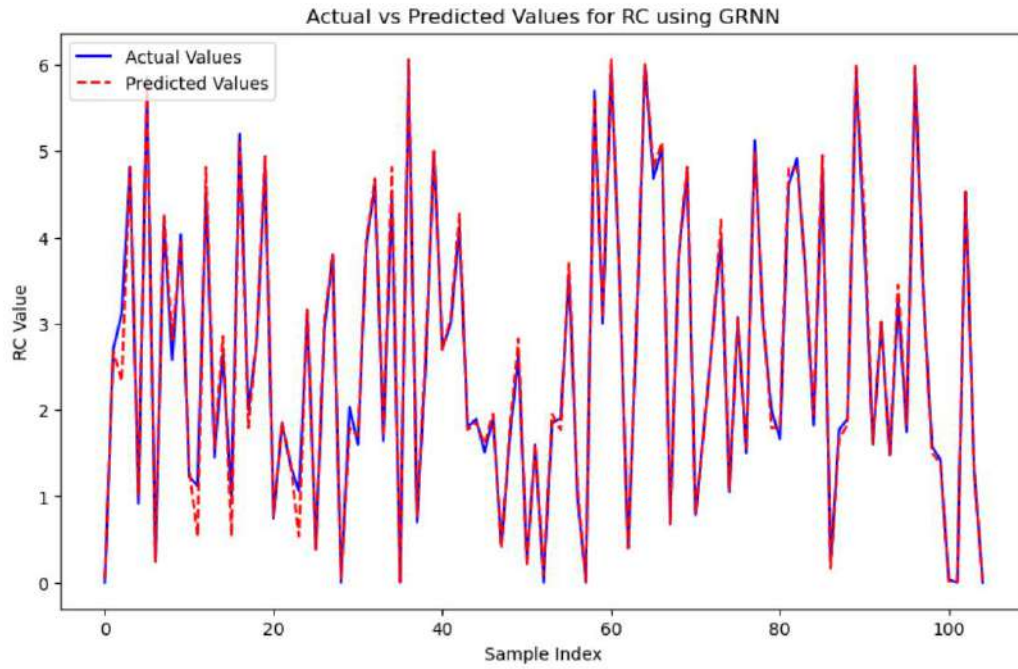


Figure (4-44): Actual and Predicted Values for GRNN S1 direct 2.

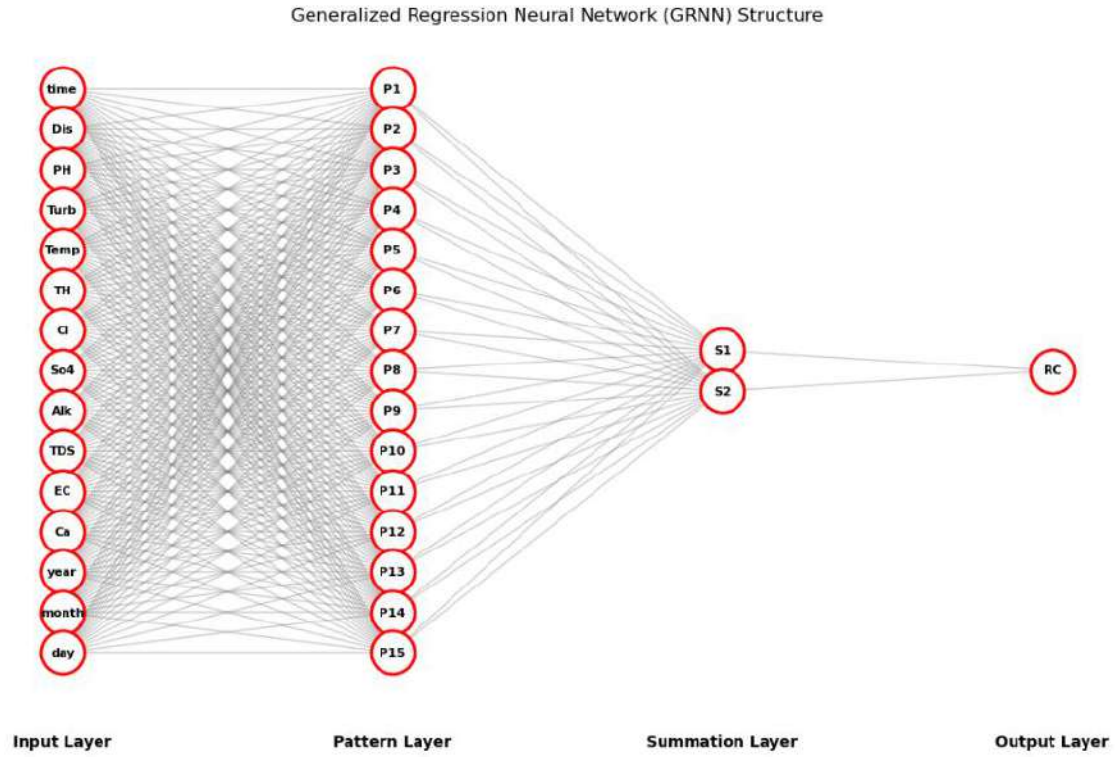


Figure (4-45): GRNN structure Scenario 1 direct 2.

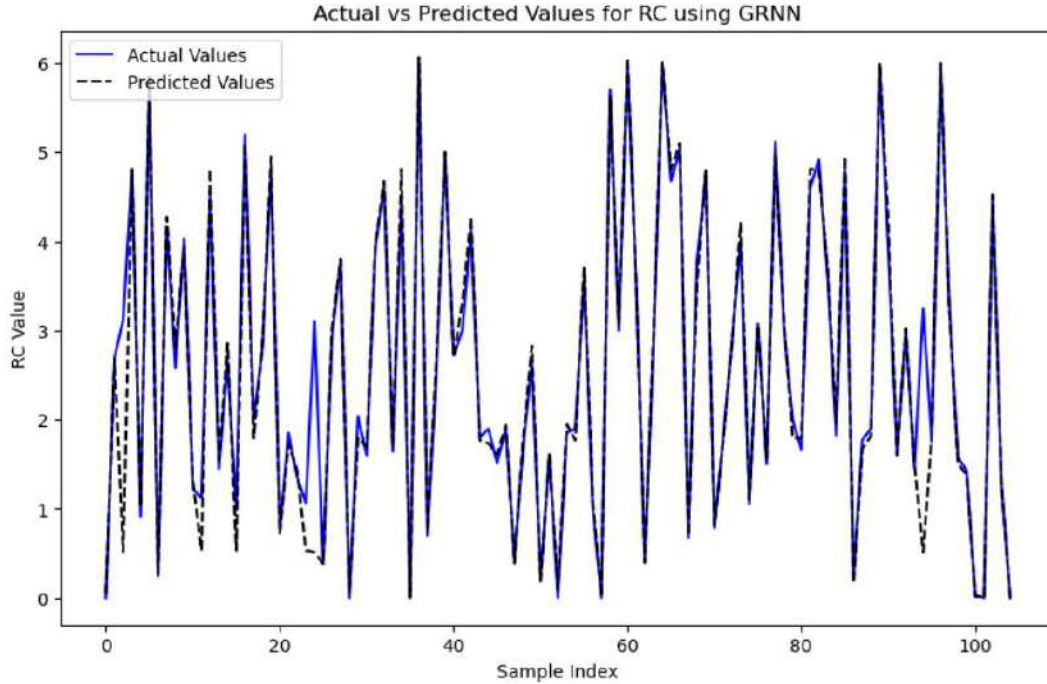


Figure (4-46): Actual and predicted values for GRNN S2 direct 2.

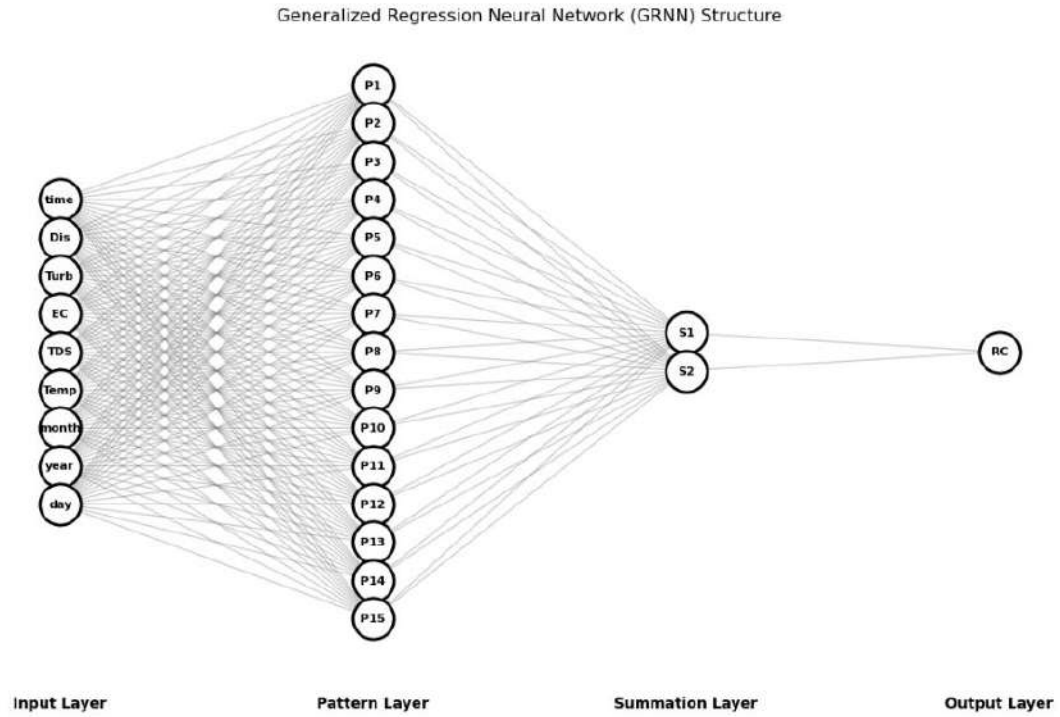


Figure (4-47): GRNN structure Scenario 2 direct 2.

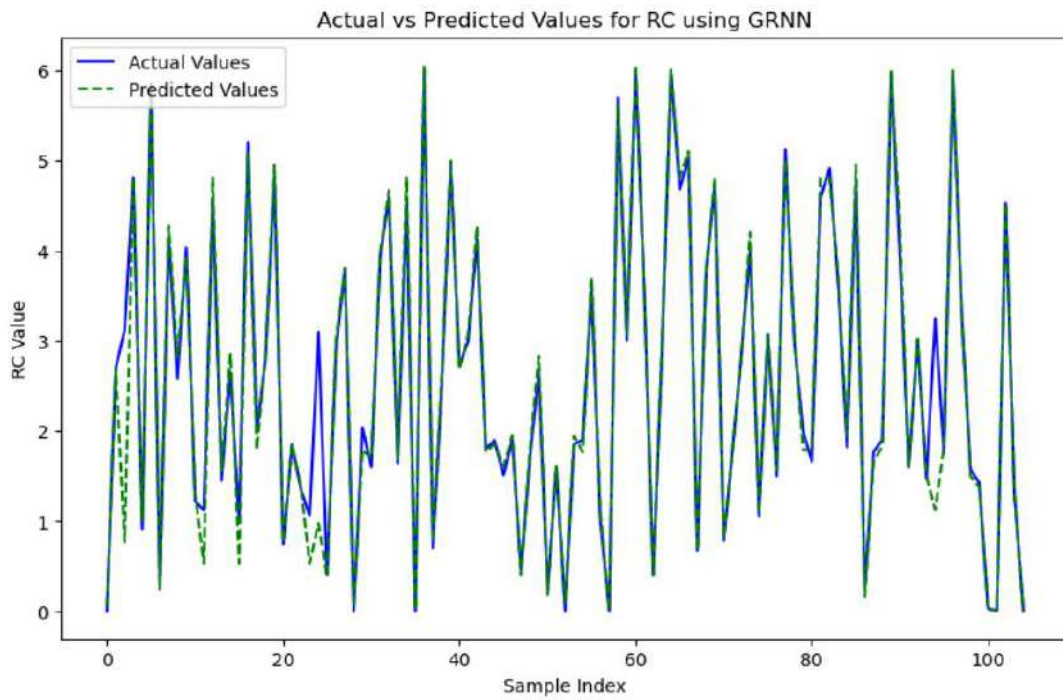


Figure (4-48): Actual and Forecasted Values for GRNN S3 direct 2.

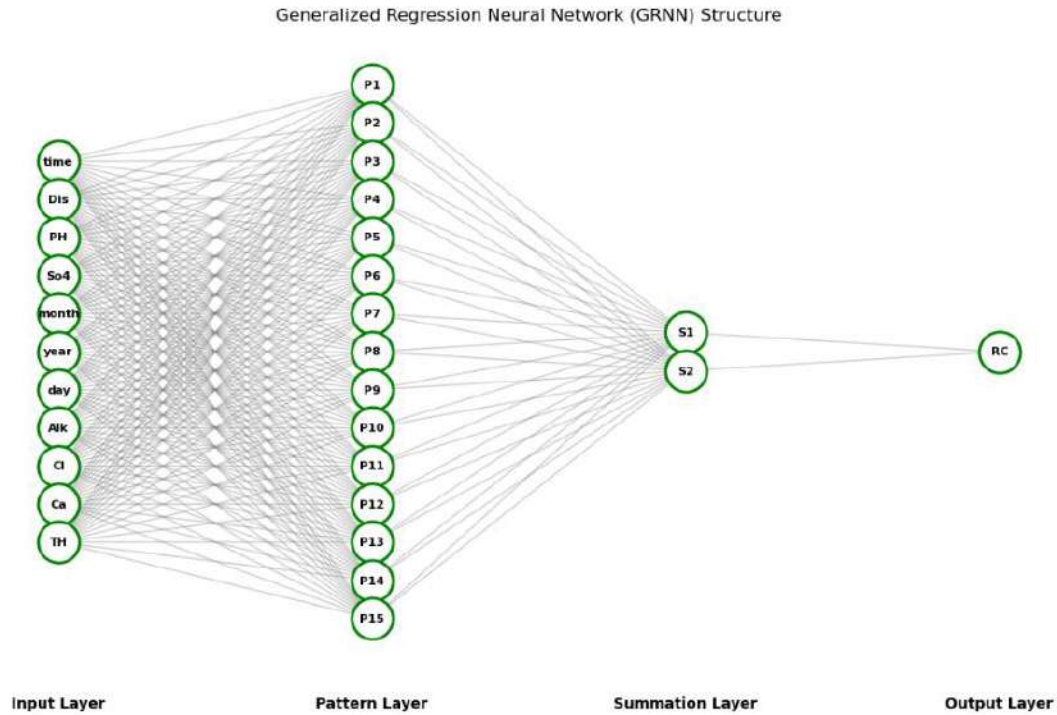


Figure (4-49): GRNN structure Scenario 3 direct 2.

4.4.6 Transformers Model

For the direct 1, the first scenario demonstrated superiority with an R-square of 0.92 as shown in Figure (4-50), closely followed by the third scenario Figure (4-52), which recorded an R-square of 0.918, exhibiting only a marginal difference from the first scenario. Then the second scenario exhibited an R-square of 0.90 Figure (4-51) (Table 4-24). (See Appendix B, Figure B.25, B.26 and B.27).

For the direct 2, the first scenario exhibited the highest possible efficiency, attaining an R-square of 0.949 in Figure (4-53), succeeded by the third scenario at 0.934 Figure (4-55), and at last, the second scenario at 0.904 Figure (4-54) (Table 4-26). (See Appendix B, Figure B.28, B.29 and B.30).

Regarding the errors, the first direction exhibited the minimal error in the first scenario. Then, the third scenario presents an error with $MSE = 0.072$, $MAE = 0.186$, and $RMSE = 0.268$. And the highest error values occur in the second scenario, with $MSE=0.086$, $MAE=0.211$, and $RMSE=0.293$.

Both directions generally agreed on the supremacy of the first scenario, which includes all features together. Although they are minimal weight, they enable the algorithm to forecast the RC and to recognize its features within the network, where the model's efficacy resides in controlling its complex relationships among the various data points. And the weakest performance among them is the scenario that includes dataset 2, as the absence of dataset 3 reduces the model's efficacy. As for the third scenario, it possessed slightly inferior performance relative to the first scenario. This gives the impression that dataset 3 such as SO_4^{-2} , Alk, Cl, Ca, TH, and pH significantly impact the behavior of RC, in terms of its reaction speed within the network. The observable differences between the scenarios in both directions are due to the variance in data quality in each direction.

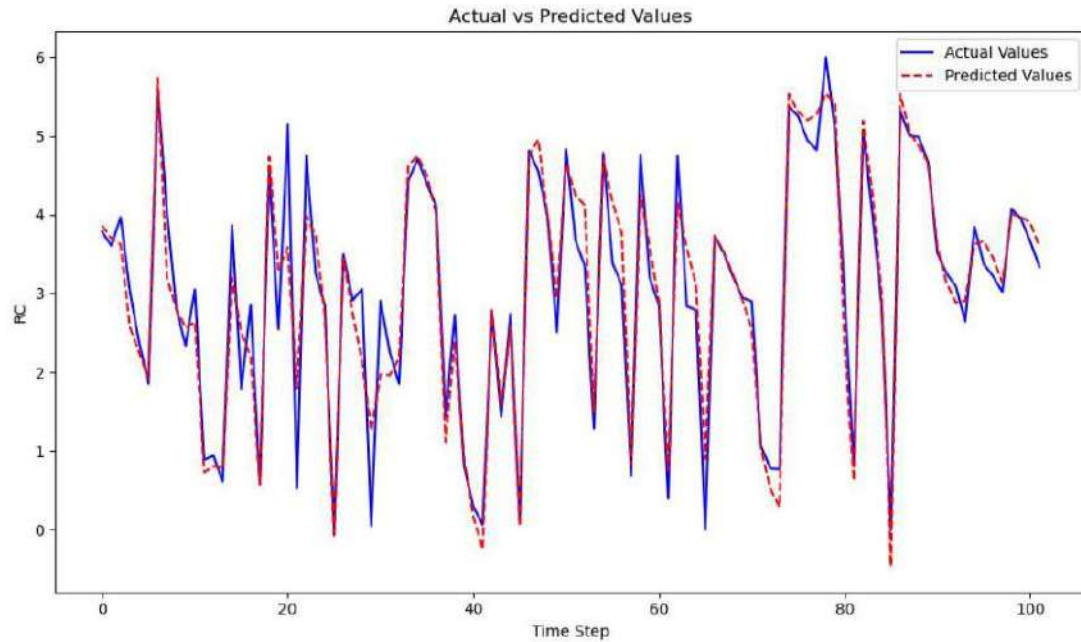
The model's peak performance occurred in the first scenario of the second direction, reaching about 95%, a significant percentage with potential for enhancement. This demonstrates that the Transformer model proved its efficacy and success with the data in the current study. Because of the absence of prior research in this particular field, there is no comparison with other studies.

Table (4-23): the hyperparameters for Transformer direct 1.

Transformer _D1	Scenario 1	Scenario 2	Scenario 3
Model Dim	64	128	128
Num Heads	4	4	2
Num Layers	2	4	2
Learning Rate	0.001	0.0001	0.001

Table (4-24): Performance metric for Transformer direct 1.

Transformer _D1	Scenario 1	Scenario 2	Scenario 3
R ²	0.920	0.902	0.918
MSE	0.071	0.086	0.072
MAE	0.190	0.211	0.186
RMSE	0.264	0.293	0.268

**Figure (4-50):** Actual and Forecasted Values by Transformer S1 direct 1.

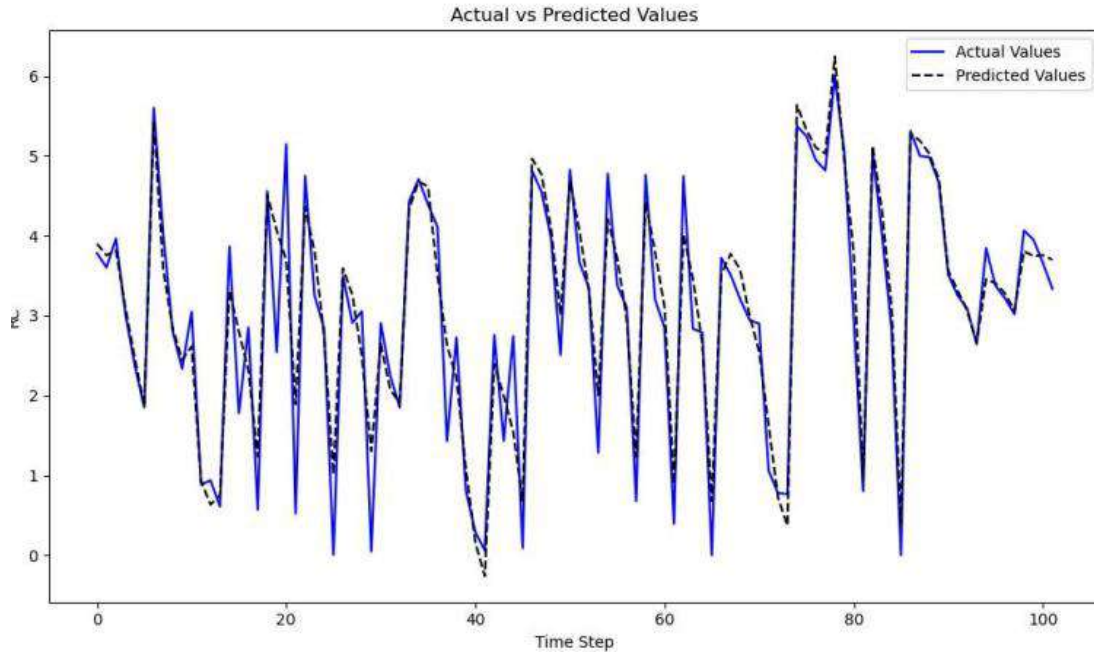


Figure (4-51): Actual and Forecasted Values by Transformer S2 direct 1.

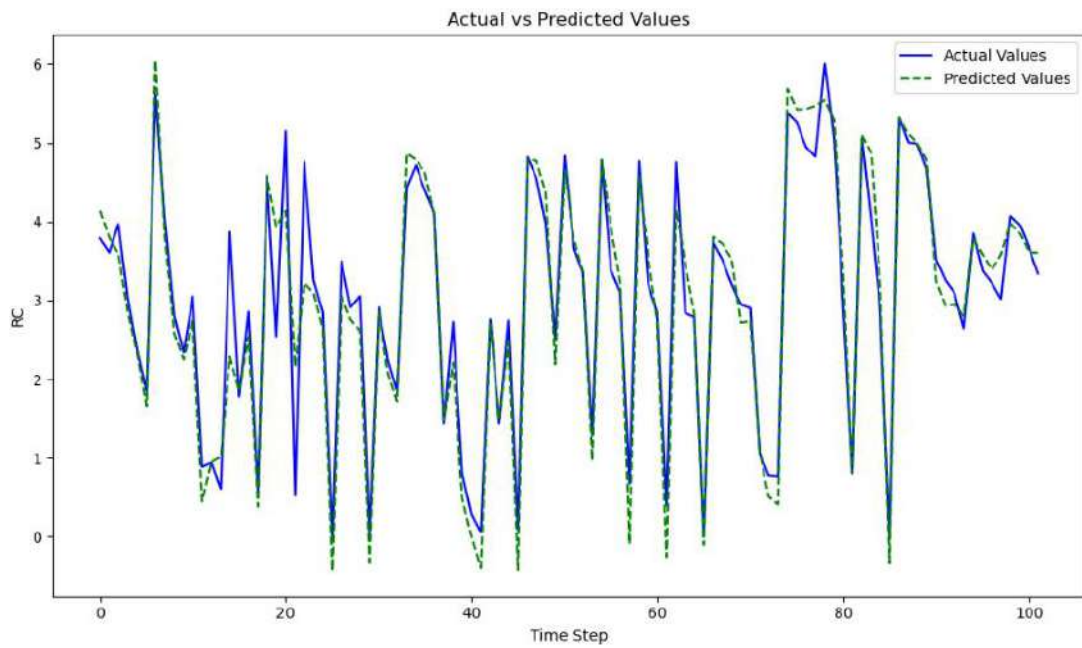


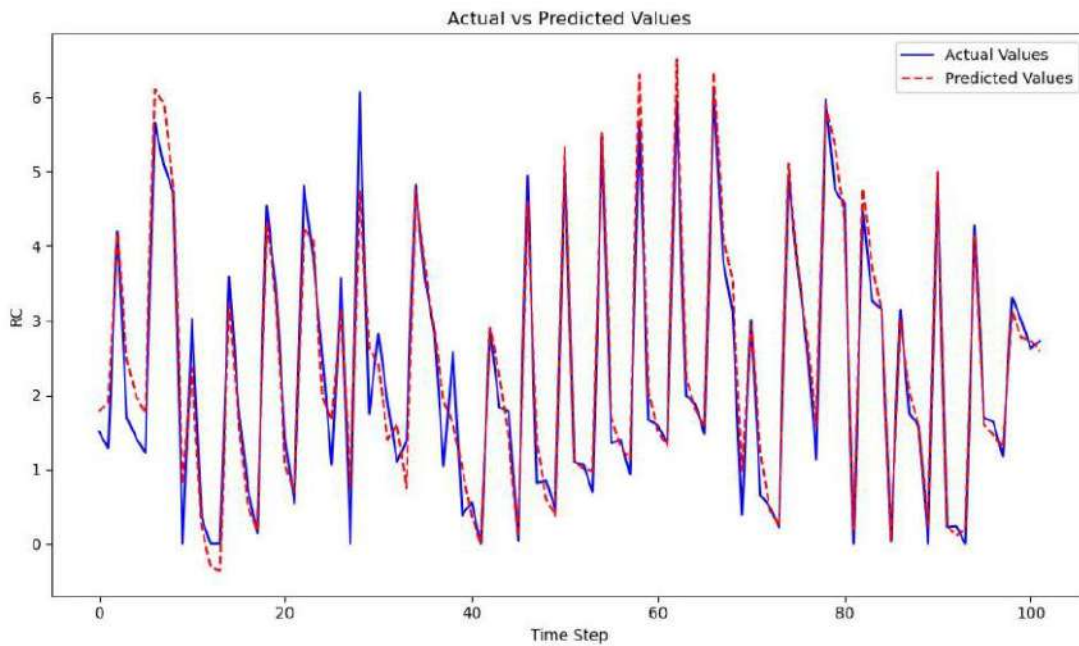
Figure (4-52): Actual and Forecasted Values by Transformer S3 direct 1.

Table (4-25): the hyperparameters for Transformer direct 2.

Transformer_D2	Scenario 1	Scenario 2	Scenario 3
Model Dim	128	128	128
Num Heads	2	2	2
Num Layers	4	4	4
Learning Rate	0.0001	0.0001	0.0001

Table (4-26): Performance metric for Transformer direct 2.

Transformer _D2	Scenario 1	Scenario 2	Scenario 3
R ²	0.949	0.904	0.934
MSE	0.053	0.101	0.069
MAE	0.175	0.238	0.188
RMSE	0.231	0.318	0.263

**Figure (4-53):** Actual and Forecasted Values by Transformer S1 direct 2.

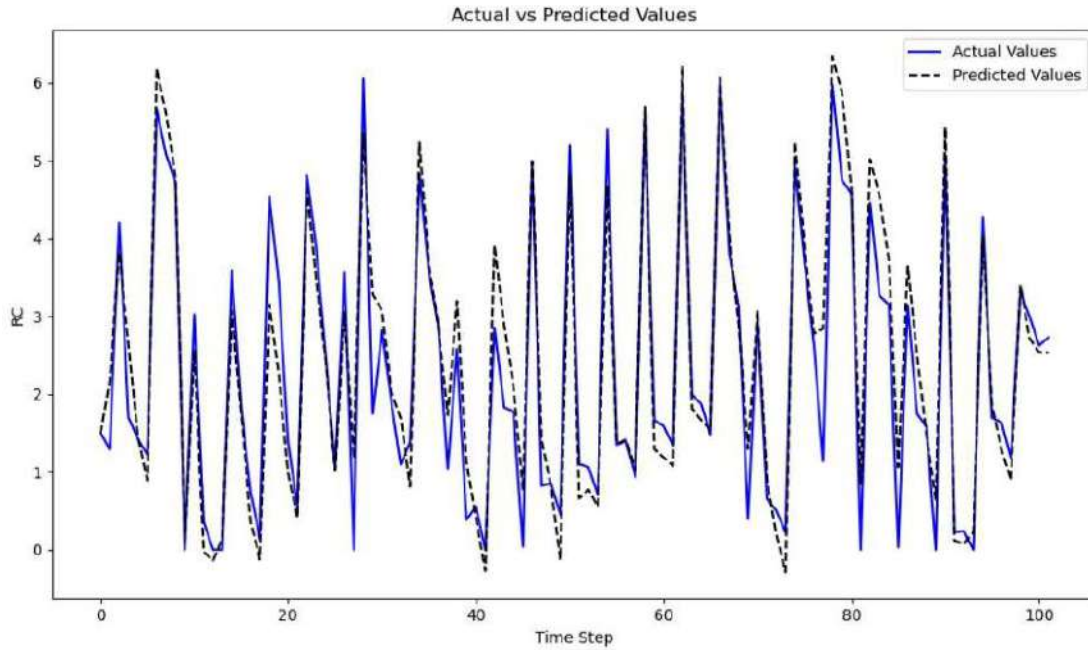


Figure (4-54): Actual and Forecasted Values by Transformer S2 direct 2.

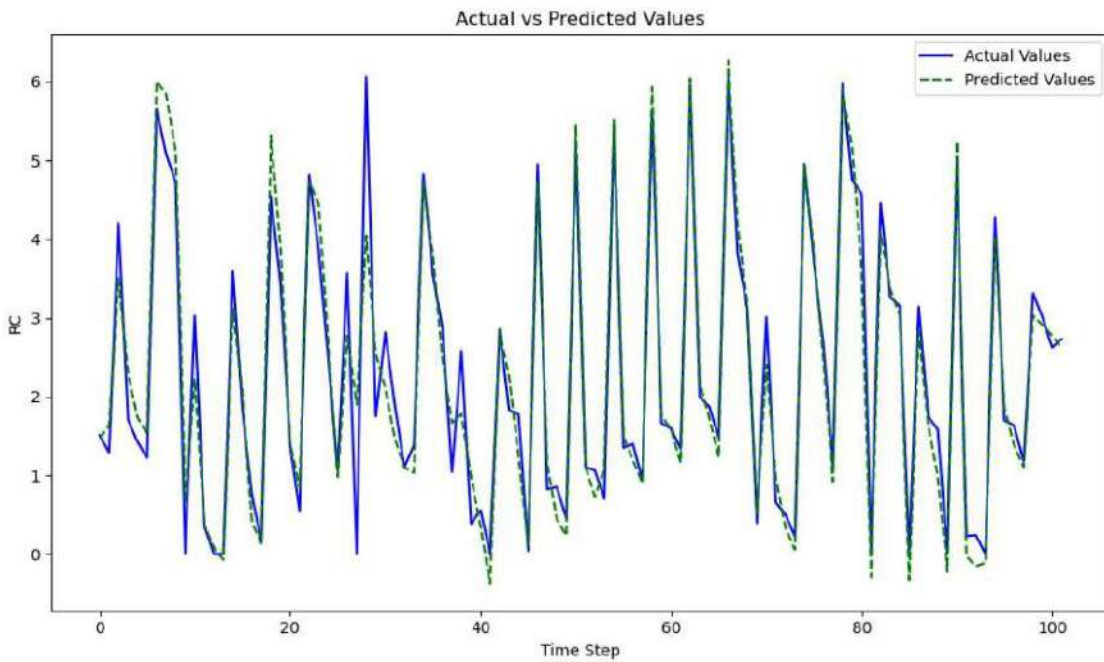


Figure (4-55): Actual and Forecasted Values by Transformer S3 direct 2.

4.5 Comparison of the models performance

The models have been evaluated, and the optimal model was identified based on the maximum R^2 and the minimum MSE, MAE, and RMSE errors. Due to

the similarity of R^2 values across scenarios, model comparison was based on error metrics, with MSE adopted as the main evaluation criterion to identify the optimal model. The disparity in accuracy and performance measures between the models occurs from their inherent properties in data processing and hyperparameter adjustment. This results from the algorithm's efficacy in prediction and its ability to handle this type of information. Almost all scenarios approved that dataset 2 alone are inadequate for training an appropriate model, and that dataset 3 are highly significant for the model's performance.

For the direct 1 Figure (4.56) presents a comparison of the five models for (S1). It is observed that the lowest MSE is achieved by the GRNN model with an MSE of 0.009, indicating superior performance. The Transformer model follows with an MSE of 0.071, while XGBoost and RF record values of 0.103 and 0.205, respectively. The weakest model in this scenario is the MLP, which reaches the highest error value of 0.242. And Figure (4.57) illustrates the model comparison for the S2 for the same direction. The results show that the Transformer model achieves the lowest error with an MSE of 0.086, followed by the GRNN with 0.132. The XGBoost model ranks third with 0.161, while the RF records 0.263. Once again, the MLP demonstrates the poorest performance, with the error reaching its maximum value of 0.625. Although S2 generally exhibits weaker performance compared to S1, the Transformer and GRNN models remain the most effective among the tested approaches. In Figure (4.58) shows the comparison of the models under the first direction for S3. The results highlight that the GRNN model achieves the lowest error, with an MSE of 0.01, which is remarkably low and reflects excellent model performance. The Transformer model follows with an MSE

of 0.072, while the XGBoost records 0.18. The RF model shows a slightly higher error of 0.197. The poorest-performing model in this scenario is once again the MLP, with an error reaching 0.45, which is considered high and thus indicates the weakest performance among the tested models.

Regarded to the direct two where Figure (4.59) presents the comparison of the models under the second direction for Scenario 1. The results indicate that the GRNN model achieves the lowest MSE value (0.024), followed closely by the Transformer model (0.053). The XGBoost model ranks third with an MSE of 0.102, followed by the MLP with 0.132. The weakest performance in this scenario is observed in the RF model, which records the highest error of 0.199. And Figure (4.60) illustrates the performance of the models in S2. In this case, the Transformer model demonstrates the lowest error with an MSE of 0.101, followed by the GRNN with 0.219. The XGBoost model comes next with 0.335, while both the MLP (0.586) and RF (0.595) show the highest error values, reflecting the weakest performance in this scenario. In Figure (4.61) shows the comparison of the models under Scenario 3 for the second direction. Once again, the Transformer model achieves the lowest error (0.069), confirming its stable and consistent performance direct two. The XGBoost model ranks second with an MSE of 0.105, followed by the MLP with 0.117. The GRNN records a higher error of 0.156, while the RF exhibits the weakest performance with the highest error value of 0.259.

The GRNN achieved the lowest error number, with an MSE of 0.009, demonstrating its higher predictive accuracy in specific cases. It frequently appeared as the superior model in 3 of the 6 scenarios in both directions. And in the other three scenarios, the Transformer excelled, especially in D2. However, GRNN performance varied noticeably across directions. In contrast

to the Transformer, while not achieving the lowest possible error, it produced the most consistent results, with MSE values ranging closely between 0.053 and 0.101, indicating a higher level of stability and strength. As a result, the two models are competing for the first rank in overall performance. As the research proves, GRNN is a high-performance model, with the neural network successfully identifying complicated patterns in the data. and the researchers prove that the GRNN model's success can be attributed to its simplicity, rapid training efficiency, fixed network architecture that required less improvement, and its proficiency in modeling nonlinear interactions. GRNN can approximate any randomized function based on historical information. **(Bowden et al., 2006).**

And for the XGBoost model MSE ranging between 0.102 and 0.335 where, the XGBoost model achieved third position and demonstrated notable accepted in some scenarios. And the model has proved its good in predicting RC levels in DW it attributed to its strength and readability in complex systems **(Ali et al., 2023)**. And handling data pertaining to complex water quality. this results from its approach to data management, which involves constructing decision trees that continually learn from previous errors. This can be expected in enhancing the model's accuracy and minimize potential errors, identified by superior generalization performance **(Zhou et al., 2019)**. As for the GRNN's superiority over XGBoost, it is attributed to its heightened sensitivity to noise in the data; in contrast, the GRNN is capable of handling it. Due to insufficient research to predict RC in this algorithm, there is still potential to enhance its reliability and accuracy. In terms of the RF model, The RF model MSE ranging between 0.197 and 0.595, It demonstrated

acceptable performance in some scenarios and unacceptable performance in others.

As for the MLP model, it is the least efficient, ranging between 0.117 and 0.625. Although it's good in S1 and S3 in direct 2, it is not favored due to its inconsistent and unstable behavior in various scenarios, making it unreliable. Where the variation in performance across each case is explained because the model is a multi-layer neural network, needing strong hyperparameter tuning to align with the nature of the data on every case. In contrast to the adaptability of the GRNN model.

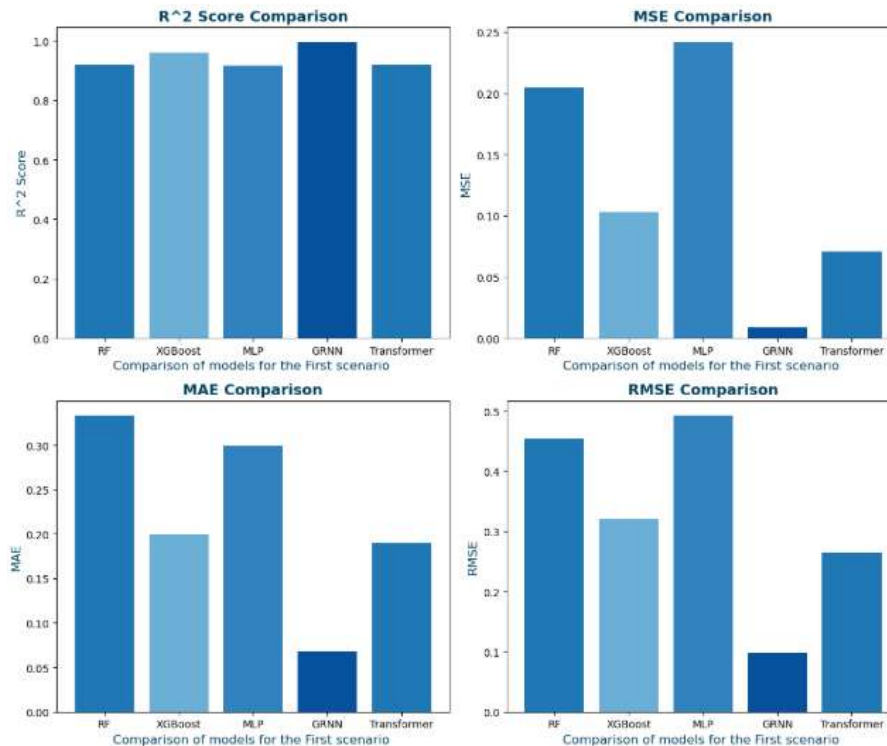


Figure (4-56): The bar chart reflects the variation of the models in S1, direct 1.

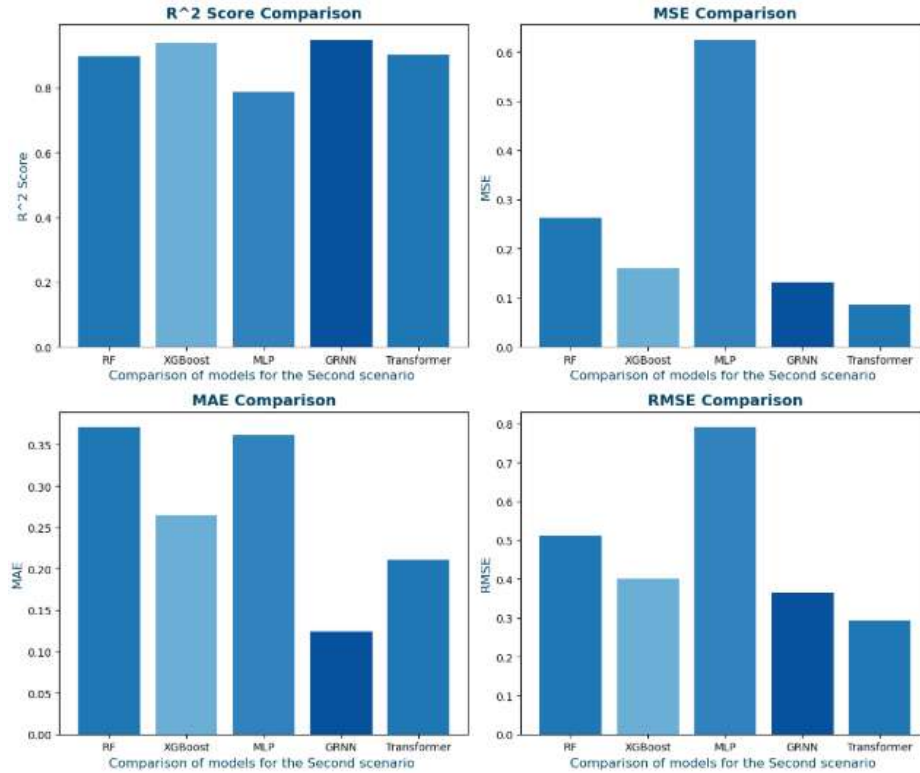


Figure (4-57): The bar chart reflects the variation of the models in the S2, direct 1.

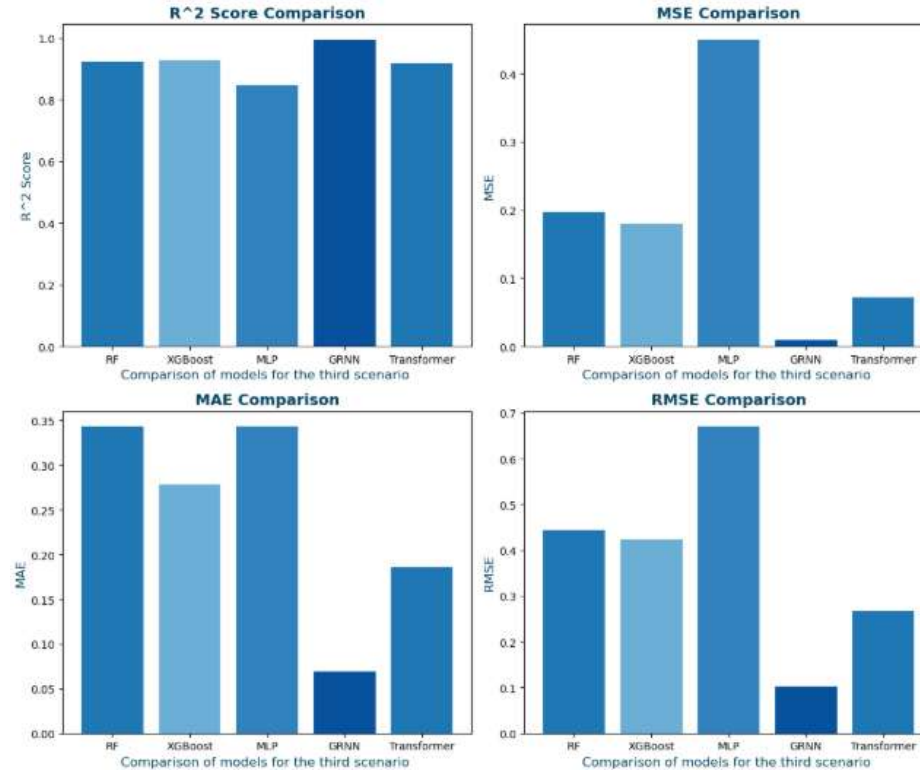


Figure (4-58): The bar chart reflects the variation of the models in the S3, direct 1.

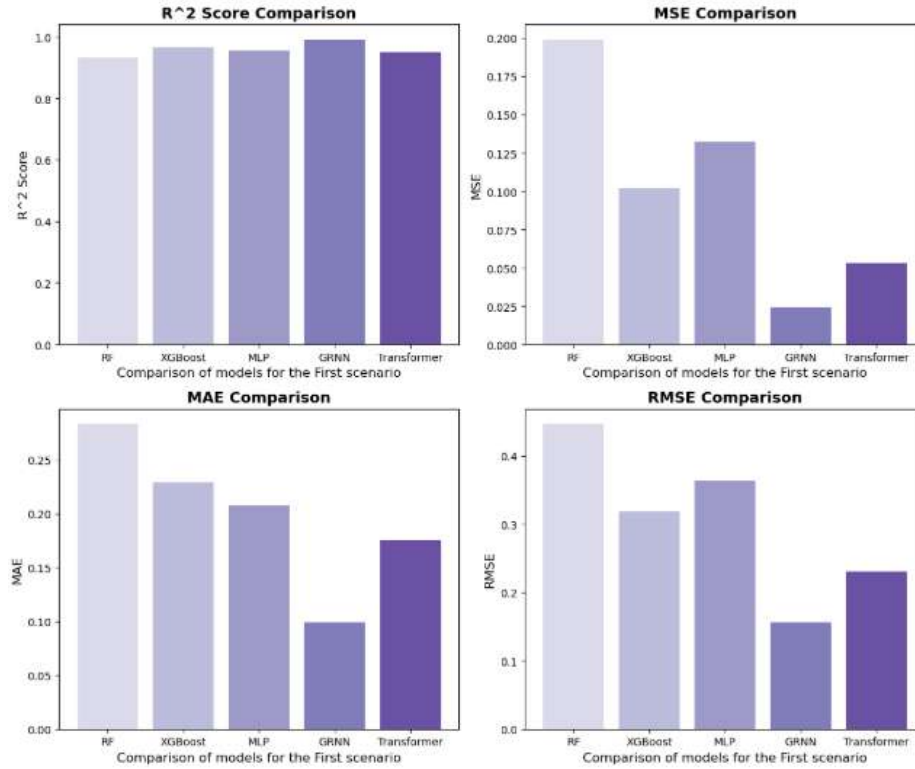


Figure (4-59): Comparison of models of the S1 direct 2.

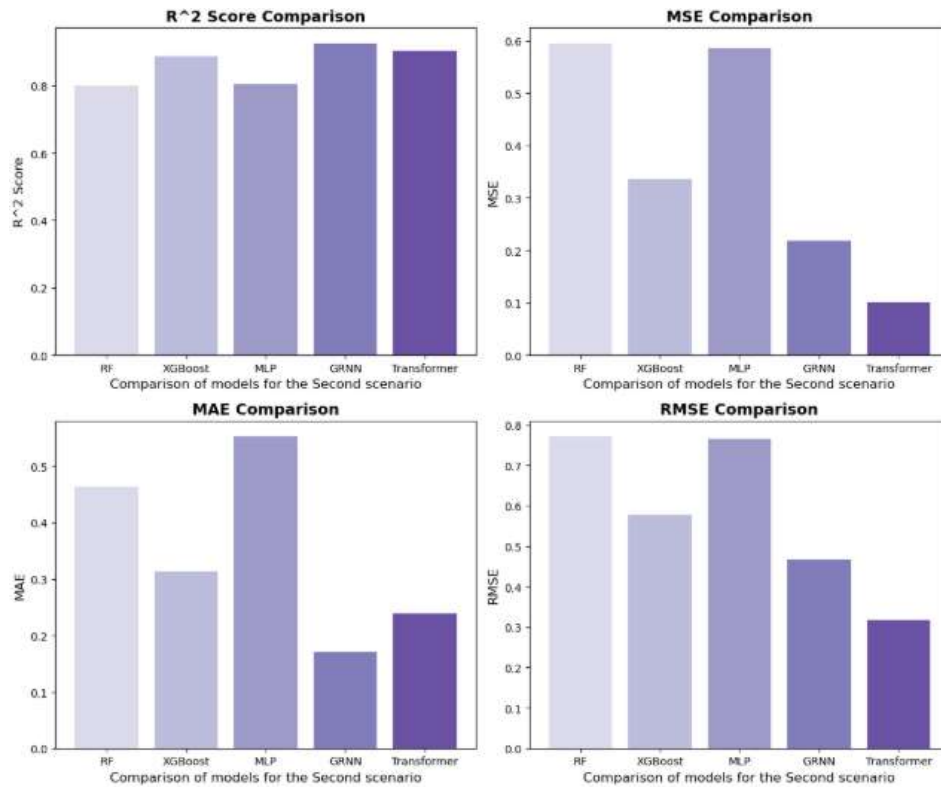


Figure (4-60): Comparison of models of the S2 direct 2.

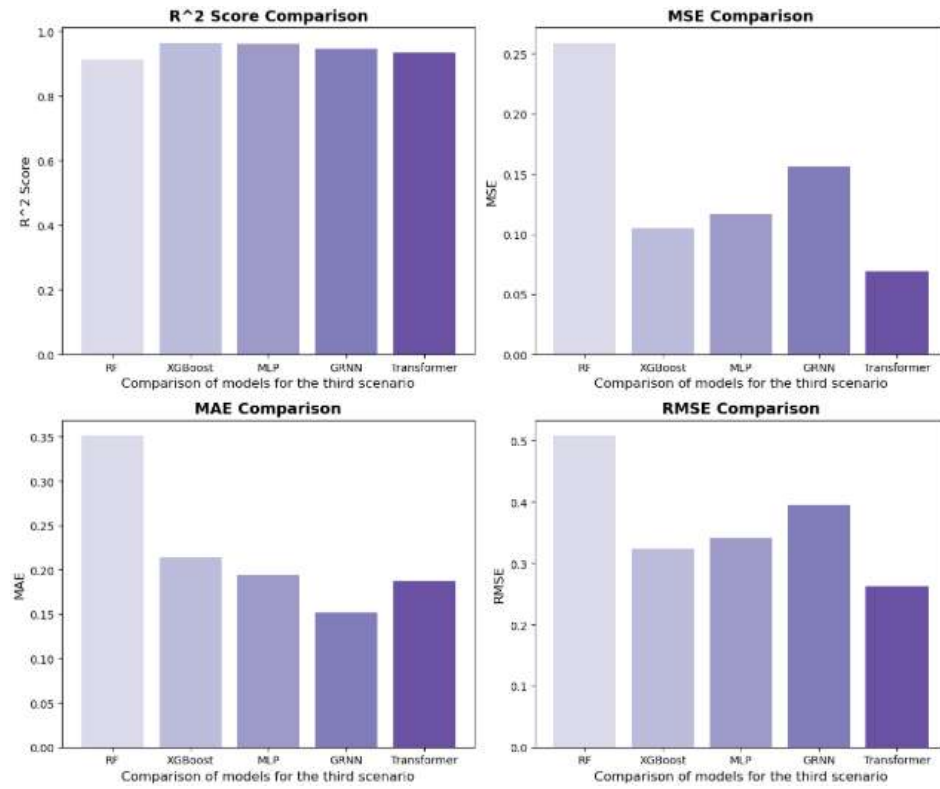


Figure (4-61): Comparison of models of the S3 direct 2.

Chapter Five: Conclusions and Recommendations

5.1 Introduction

The primary goal of this chapter is to briefly summarize the results for practical and programming aspects clearly and concisely. Along with recommendations for the research's practical application and ideas that may help progress future, more advanced research.

5.2 Conclusion

The most significant conclusion of the research is provided in the following points:

1. The study compared several ML models (RF, XGBoost, GRNN, MLP, Transformer) for predicting RC in the Karbala DWDN.
2. Water quality analysis revealed that all variables complied with WHO and Iraqi DW standards, with the exception of turbidity and RC, which showed a significant increase due to The observed rise in turbidity was associated with the plant's ineffectiveness in eliminating suspended particles, attributed to deficiencies in filters, sedimentation tanks, and the substandard quality of alum. Increased turbidity not only causes network blockage but also diminishes disinfection efficacy.
3. Also, RC had been experiencing an overdose, which might lead to issues such as network degradation over time.
4. The models were evaluated, demonstrating that the GRNN and Transformer are the most effective predictive models due to its effective capture of complicated relationships,
5. While XGBoost ranked second, performing reasonably acceptably in some scenarios and able to grow, whereas RF showed variable

performance and MLP was the least reliable due to instability and high sensitivity to hyperparameters.

6. To the best of our knowledge, this research is the first to apply AI models for forecasting RC in Karbala. In addition, it is among the first to employ XGBoost and Transformer algorithms for predicting RC within the network, which may contribute to advancing the management of DW quality.
7. The findings of this research supported the potential of incorporating these models into the automation system at the plant and network, facilitating the model to predictions in real-time.

5.3 Recommendations

1. Based on the obtained results, it is beneficial to address the turbidity issue by reconsidering the plant's maintenance, continuously monitoring the filters, and ensuring the quality of the alum for enhanced operational efficiency.
2. The results from comparing predictive models can be used to create a better model and connect it to an automated monitoring system, which provides up-to-date information about the distribution network for DW and forecasts chlorine levels before customers access it at strategic locations. This project necessitates collaboration with pertinent authorities to automate the system for monitoring water quality, hence facilitating the obtaining of large amounts of data for researchers to analyze more clearly and comprehensively. This could reduce the efforts exerted by employees to conduct laboratory testing.

3. Future studies may utilize larger datasets, enabling the model to recognize all data patterns, to assess model stability, and can employ other algorithms. It may even include seasonal cases ensuring the model's powerful ability for future predictions with new data successfully.
4. Feature engineering can be used to directly understand the influence of each variable. In addition, to using physical and chemical factors as datasets to better understand their impact on the model.
5. It is necessary to identify critical locations within the network as monitoring stations, which might function as chlorination injection stations to enhance chlorine levels in the event of a sudden drop.
6. In the end, this research serves as an introduction to the field of using AI technology to develop monitoring methods within the DWDN in Karbala. It is hoped it will be taken seriously and become a foundation for further study that advances this work, considering the significance of integrating smart technology to deal with these challenges, as the issue of chlorine is highly sensitive and directly linked to human health.

REFERENCES

- Abbas, A.H. (2017) 'Studying of Residual Chlorine Concentration within Water Supply Distribution System in Samarra City - Iraq', (September 2011), pp. 1–6.
- Abdulwahid, A.H. (2020) 'IoT Based Water Quality Monitoring System for Rural Areas', *9th International Conference on Renewable Energy Research and Applications, ICRERA 2020*, pp. 279–282. Available at: <https://doi.org/10.1109/ICRERA49962.2020.9242798>.
- Abed, Z.H. and Khudair, K.M. (2023) 'Investigation of high-turbidity tap water problem in Najaf governorate/middle of Iraq', *Open Engineering*, 13(1), pp. 1–12. Available at: <https://doi.org/10.1515/eng-2022-0425>.
- Ahad, N.A. and Yahaya, S.S.S. (2014) 'Sensitivity analysis of Welch's t -test', *AIP Conference Proceedings*, 1605, pp. 888–893. Available at: <https://doi.org/10.1063/1.4887707>.
- Ainapure, B. *et al.* (2023) 'Drinking water potability prediction using machine learning approaches: a case study of Indian rivers', *Water Practice and Technology*, 18(12), pp. 3004–3020. Available at: <https://doi.org/10.2166/wpt.2023.202>.
- Al-Hussein neighborhood water project* (2023) *Holy Karbala Governorate*. Available at: <https://karbala.gov.iq/upload/63efb06949013.jpg>.
- Al-Otoun, F. *et al.* (2016) 'Disinfection by-products of chlorine dioxide (chlorite, chlorate, and trihalomethanes): Occurrence in drinking water in Qatar', *Chemosphere*, 164, pp. 649–656. Available at: <https://doi.org/10.1016/j.chemosphere.2016.09.008>.
- Aldahdouh, A. (2017) 'Does Artificial Neural Network Support Connectivism's Assumptions? Can Connectivism Theory Explain How Millennials Learn? View project', *International Journal of Instructional Technology and Distance Learning*, 14(3). Available at: <https://www.researchgate.net/publication/317905136>.
- Ali, Z.A. *et al.* (2023) 'eXtreme gradient boosting algorithm with machine learning: A review', *Academic Journal of Nawroz University*, 12(2), pp. 320–334.
- and Eddy, I.M. *et al.* (2002) 'Wastewater engineering: treatment and reuse'. McGraw-Hill Science/Engineering/Math.
- Anne, A. *et al.* (2023) 'Impact of the Condition of Drinking Water Supply Networks on the Quality of Water Intended for Consumption : The Case of the

Network in the Commune of Daloa (Central West of Côte d ' Ivoire)', pp. 677–691. Available at: <https://doi.org/10.4236/jwarp.2023.1512036>.

APHA, A.P.H.A. (2017) *Standard Methods for the Examination of Water and Wastewater*. 23rd edn. American Public Health Association.

Batista, G. de S. *et al.* (2024) 'Modeling the decay of free residual chlorine in water distribution networks in Brazilian rural communities using artificial neural network'. Available at: <https://www.sciencedirect.com/science/article/abs/pii/S2214714424005440>.

Belcaid, A.E., Taghlabi, F. and Tounsi, A. (2023) 'Chlorine Decay Modeling in a Water Distribution Network in Mohammedia City, Morocco', *Ecological Engineering and Environmental Technology*, 24(2), pp. 269–278. Available at: <https://doi.org/10.12912/27197050/157538>.

Biau, G. and Scornet, E. (2016) 'A random forest guided tour', *Test*, 25(2), pp. 197–227. Available at: <https://doi.org/10.1007/s11749-016-0481-7>.

EL Bilali, A. *et al.* (2021) 'An integrated approach based on Gaussian noises-based data augmentation method and AdaBoost model to predict faecal coliforms in rivers with small dataset', *Journal of Hydrology*, 599(November 2020), p. 126510. Available at: <https://doi.org/10.1016/j.jhydrol.2021.126510>.

Bolton, J.R. and Cotton, C.A. (2011) *The ultraviolet disinfection handbook*. American Water Works Association.

Bonton, A. *et al.* (2012) 'Comparative life cycle assessment of water treatment plants', *Desalination*, 284, pp. 42–54. Available at: <https://doi.org/10.1016/j.desal.2011.08.035>.

Bowden, Gavin J. *et al.* (2006) 'Forecasting chlorine residuals in a water distribution system using a general regression neural network', *Mathematical and Computer Modelling*, 44(5–6), pp. 469–484. Available at: <https://doi.org/10.1016/j.mcm.2006.01.006>.

Bowden, Gavin J *et al.* (2006) 'Forecasting chlorine residuals in a water distribution system using a general regression neural network', 44, pp. 469–484. Available at: <https://doi.org/10.1016/j.mcm.2006.01.006>.

Brander, N. (2003) 'Drinking water in schools.', *Nursing times*, 99(1), pp. 50–51. Available at: <https://doi.org/10.4135/9781452276250.n68>.

Al Chalabi, A.S. (2020) 'Assessment of drinking water quality and the efficiency of the Al-Buradieiah water treatment plant in Basra City', *Nature Environment and Pollution Technology*, 19(3), pp. 1057–1065. Available at:

<https://doi.org/10.46488/NEPT.2020.v19i03.016>.

Chelliah, P.R., Sakthivel, U. and Nagarajan, S. (2022) *Applied Learning Algorithms for Intelligent IoT*. Edited by 1st Edition. CRC Press / Auerbach Publications.

Chen, T. and Guestrin, C. (2016) ‘XGBoost: A scalable tree boosting system’, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-Aug, pp. 785–794. Available at: <https://doi.org/10.1145/2939672.2939785>.

Chicco, D., Warrens, M.J. and Jurman, G. (2021) ‘The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation’, *PeerJ Computer Science*, 7, pp. 1–24. Available at: <https://doi.org/10.7717/PEERJ-CS.623>.

Chinnappan, C.V. *et al.* (2023) ‘IoT-Enabled Chlorine Level Assessment and Prediction in Water Monitoring System Using Machine Learning’, *Electronics (Switzerland)*, 12(6). Available at: <https://doi.org/10.3390/electronics12061458>.

Cigizoglu, H.K. (2005) ‘Generalized regression neural network in monthly flow forecasting’, *Civil Engineering and Environmental Systems*, 22(2), pp. 71–81. Available at: <https://doi.org/10.1080/10286600500126256>.

Cristina, S. (2023) *The Transformer Attention Mechanism*. Available at: <https://machinelearningmastery.com/the-transformer-attention-mechanism/>.

Crittenden, J.C. *et al.* (2023) ‘Stantec’s Water Treatment. Principles and Design.pdf’, p. 1952.

Darren George, P.M. (2016) *IMB SPSS Statistic 23 Step by Step*.

DEFRA (2018) ‘Water Treatment Processes - Drinking Water Inspectorate’, *Water treatment process*, pp. 1–36.

Engelbrecht, A.P. (2007) *Computational intelligence: an introduction*. John Wiley & Sons.

EPA (2013) *Water Treatment Manual : Disinfection*.

Fardami, A.Y., Mamuda, B. and Kangiwa, I.A. (2019) ‘Bacteriological Analysis of Drinking Water in Zamfara North Senatorial District, Nigeria’, *Microbiology Research Journal International*, (May 2019), pp. 1–10. Available at: <https://doi.org/10.9734/mrji/2019/v27i530110>.

Fesaghandis, G.S. *et al.* (2017) ‘Comparison of multilayer perceptron and

radial basis function in predicting success of new product development’, *Eng. Technol. Appl. Sci. Res.*, 7.

FireFortySix (2020) *Anaconda-logo*. Available at: https://i0.wp.com/firefortysix.com/wp-content/uploads/2020/08/Anaconda-logo-16_9-1.jpg?fit=1200%2C675&ssl=1.

Galindo Padilha, G.A. *et al.* (2022) ‘Transformer-Based Hybrid Forecasting Model for Multivariate Renewable Energy’, *Applied Sciences (Switzerland)*, 12(21). Available at: <https://doi.org/10.3390/app122110985>.

García-Ávila, F. *et al.* (2021) ‘Modeling of residual chlorine in a drinking water network in times of pandemic of the SARS-CoV-2 (COVID-19)’, *Sustainable Environment Research*, 31(1). Available at: <https://doi.org/10.1186/s42834-021-00084-w>.

Gelete, G. *et al.* (2020) ‘Evaluating disinfection techniques of water treatment’, *Desalination and Water Treatment*, 177(May 2019), pp. 408–415. Available at: <https://doi.org/10.5004/dwt.2020.25070>.

Gillioz, A. *et al.* (2020) ‘Overview of the Transformer-based Models for NLP Tasks’, in *2020 15th Conference on computer science and information systems (FedCSIS)*. IEEE, pp. 179–183.

Gupta, A. (2018) ‘Introduction to deep learning: Part 1’, *Chemical Engineering Progress*, 114(6), pp. 22–29.

Halliday, D., Resnick, R. and Walker, J. (2014) *Fundamentals of Physics Halliday & resnick 10ed*, Wiley.

Hameed, Y. *et al.* (2018) ‘Concentration of Residual Chlorine and Its Health Effects on The Drinking Water of The Kirkuk City’, *Kafkas University Institute of Natural and Applied Science Journal*, 1(1), pp. 29–37. Available at: <https://dergipark.org.tr/download/article-file/509504>.

Hao, R. *et al.* (2017) ‘Effect of water chemistry on disinfection by-product formation in the complex surface water system’, *Chemosphere*, 172, pp. 384–391.

Hari (2024) *How to launch Jupyter Notebook from any folder*. Available at: <https://miro.medium.com/v2/resize:fit:1100/format>:

Ibraheem, S., Tuama Al-Awadi, A. and Al-Saadi, R.J.M. (2024) ‘Applying SWOT Analysis to Manage Maintenance of Municipal Water Projects in Karbala City’, *IOP Conference Series: Earth and Environmental Science*, 1374(1). Available at: <https://doi.org/10.1088/1755-1315/1374/1/012072>.

Instruments, H. (2025) *HI93703 Portable Turbidity Meter ISO Compliant*. Available at: <https://hannainst.com/hi93703-portable-turbidity-meter-iso-compliant.html>.

Investopedia (2009) *Coefficient of Determination: How to Calculate It and Interpret the Result*. Available at: <https://www.investopedia.com/terms/c/coefficient-of-determination.asp>.

iStock (2024) *Woman laboratory assistant*. Available at: <https://media.istockphoto.com/id/2159717523/vector/woman-laboratory-assistant-tests-quality-of-water-for-presence-of-microbes-and-harmful.jpg?s=612x612&w=0&k=20&c=QVf1xe6BWJoPXcH87mPvQsGLFzDN092m7qXXTCf10Xc=>.

Jebril, N., Naji, H.F. and Kuder, A.A.A. (2020) 'Bacteriological and Physiochemical Analysis of Drinking Water in Hilla City, Iraq', *SSRN Electronic Journal* [Preprint], (1998). Available at: <https://doi.org/10.2139/ssrn.3567843>.

Jeffrey Yang, Y., Haught, R.C. and Goodrich, J.A. (2009) 'Real-time contaminant detection and classification in a drinking water pipe using conventional water quality sensors: Techniques and experimental results', *Journal of Environmental Management*, 90(8), pp. 2494–2506. Available at: <https://doi.org/10.1016/j.jenvman.2009.01.021>.

Joslyn, K. (2018) 'Water Quality Factor Prediction Using Supervised Water Quality Factor Prediction Using Supervised Machine Learning Machine Learning', *REU Final Reports Research Experiences for Undergraduates on Computational Modeling Serving the City* [Preprint]. Available at: https://pdxscholar.library.pdx.edu/reu_reports.

Khor, Y., Aziz, A.R.A. and Chong, S.S. (2024) 'Recent developments and sustainability in monitoring chlorine residuals for water quality control: a critical review', *RSC Sustainability*, 2(9), pp. 2468–2485. Available at: <https://doi.org/10.1039/d4su00188e>.

Kingma, D.P. and Ba, J.L. (2015) 'Adam: A method for stochastic optimization', *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* [Preprint].

Kumar, A. and Saharia, M. (2024) *Python Essentials BT - Python for Water and Environment*. Available at: https://doi.org/10.1007/978-981-99-9408-3_3.

Kwak, S. (2023) 'Are only p-values less than 0.05 significant? A p-value greater than 0.05 is also significant!', *Journal of Lipid and Atherosclerosis*,

12(2), p. 89.

Kyritsakas, G., Speight, V. and Boxall, J. (2023) 'A data-driven model for the prediction of chlorine losses in water distribution trunk mains', *IOP Conference Series: Earth and Environmental Science*, 1136(1). Available at: <https://doi.org/10.1088/1755-1315/1136/1/012048>.

Lashgari, E., Liang, D. and Maoz, U. (2020) 'Data augmentation for deep-learning-based electroencephalography', *Journal of Neuroscience Methods*, 346(February), p. 108885. Available at: <https://doi.org/10.1016/j.jneumeth.2020.108885>.

LeChevallier, Mark W.; Au, K.-K. (2004) 'Water treatment and pathogen control: Process efficiency in achieving safe drinking-water Publisher: World Health Organization'.

Leung, M.T., Chen, A.S. and Daouk, H. (2000) 'Forecasting exchange rates using general regression neural networks', *Computers and Operations Research*, 27(11–12), pp. 1093–1110. Available at: [https://doi.org/10.1016/S0305-0548\(99\)00144-6](https://doi.org/10.1016/S0305-0548(99)00144-6).

Liu, Y., Wang, Y. and Zhang, J. (2012) 'New machine learning algorithm: Random forest', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7473 LNCS, pp. 246–252. Available at: https://doi.org/10.1007/978-3-642-34062-8_32.

M. I. Hossen Abir, T.S. (2024) 'Advancing Water Quality Assessment: Leveraging XGBoost for Precise Predictive Modeling; A Machine Learning Technique', in. IEEE.

Maharana, K., Mondal, S. and Nemade, B. (2022) 'A review: Data pre-processing and data augmentation techniques', *Global Transitions Proceedings*, 3(1), pp. 91–99. Available at: <https://doi.org/10.1016/j.gltp.2022.04.020>.

Mahato, J.K. and Gupta, S.K. (2022) 'Exploring applicability of artificial intelligence and multivariate linear regression model for prediction of trihalomethanes in drinking water', *International Journal of Environmental Science and Technology*, 19(6), pp. 5275–5288. Available at: <https://doi.org/10.1007/s13762-021-03392-1>.

Misaghi, F. *et al.* (2017) 'Guidance Manual for Compliance with the Surface Water Treatment Rules: Turbidity Provisions', *Science of the Total Environment*, 589, pp. 107–116. Available at:

<http://dx.doi.org/10.1016/j.scitotenv.2017.02.226>.

Onyutha, C. (2022) ‘Multiple Statistical Model Ensemble Predictions of Residual Chlorine in Drinking Water: Applications of Various Deep Learning and Machine Learning Algorithms’, *Journal of Environmental and Public Health*, 2022. Available at: <https://doi.org/10.1155/2022/7104752>.

Organization, W.H. (2004) *Guidelines for drinking-water quality*. World Health Organization.

Organization, W.H. (2022) *Guidelines for drinking-water quality: incorporating the first and second addenda*. World Health Organization.

Palin, A.T. (1975) ‘Water disinfection-chemical aspects and analytical control’, *Disinfection: water and wastewater*, pp. 67–89.

Perez, H. and Tah, J.H.M. (2020) ‘Improving the accuracy of convolutional neural networks by identifying and removing outlier images in datasets using t-SNE’, *Mathematics*, 8(5). Available at: <https://doi.org/10.3390/MATH8050662>.

Pichel, N., Vivar, M. and Fuentes, M. (2019) ‘The problem of drinking water access: A review of disinfection technologies with an emphasis on solar treatment methods’, *Chemosphere*, 218, pp. 1014–1030. Available at: <https://doi.org/10.1016/j.chemosphere.2018.11.205>.

Pipit Muliya, Dyah Aminatun, Sukma Septian Nasution, Tommy Hastomo, Setiana Sri Wahyuni Sitepu, T. (2020) ‘~~濟無~~No Title No Title No Title’, *Journal GEEJ*, 7(2).

Pngtree (2024) *Digital robot with laptop technology concept illustration*, Pngtree. Available at: https://png.pngtree.com/png-vector/20241117/ourmid/pngtree-digital-robot-with-laptop-technology-concept-illustration-png-image_14428848.png.

Python logo (2008) *Python Software Foundation*. Available at: <https://en.m.wikipedia.org/wiki/File:Python-logo-n>.

Python Software Foundation (2025) *The Python Tutorial*, *Python Software Foundation*. Available at: <https://docs.python.org/3/tutorial/index.html>.

Rafiqul Islam, M. (2016) ‘A Study on the TDS Level of Drinking Mineral Water in Bangladesh’, *American Journal of Applied Chemistry*, 4(5), p. 164. Available at: <https://doi.org/10.11648/j.ajac.20160405.11>.

Rammadhan, A.A., Nada, K.B. and Jassam, B.A. (2016) ‘Variation of Residual

Chlorine Concentration in Some Regions of Al_Rassafa', *Iraqi Journal of Science*, 57(2C), pp. 1436–1444.

Rawpixel (2023) *Image*. Available at: https://images.rawpixel.com/image_png_800/cHJpdmF0ZS9sci9pbWFnZXMvd2Vic2l0ZS8yMDIzLTEwL3Jhd3BpeGVsb2ZmaWNlMjBfYV9waG90b19vZl9hX3JvYm90X2hhbmRfc2hha2luZ19odW1hbl9oYW5kX2hvbF8wZDQ1OTU0Yi02ZWJjLTRlYzAtOWI3OS1kMWY5MmU2N2M1ZmMucG5n.png.

Richardson, S.M. (1989) 'Fluid mechanics.'

Ris, L. *et al.* (2019) 'Optimization for Feature Selection in Dna', 11(May), pp. 1–17.

Ruxton, G.D. (2006) 'The unequal variance t-test is an underused alternative to Student's t-test and the Mann-Whitney U test', *Behavioral Ecology*, 17(4), pp. 688–690. Available at: <https://doi.org/10.1093/beheco/ark016>.

Saikureti, R. (2020) *History of deep learning*.

Sakran, T. *et al.* (2019) 'Monitoring water quality parameters in Egyptian tap water', *Polish Journal of Environmental Studies*, 28(4), pp. 2815–2821. Available at: <https://doi.org/10.15244/pjoes/92820>.

De Santi, M. *et al.* (2021) 'Forecasting point-of-consumption chlorine residual in refugee settlements using ensembles of artificial neural networks', *npj Clean Water*, 4(1), pp. 1–16. Available at: <https://doi.org/10.1038/s41545-021-00125-2>.

Saunier, B.M. and Selleck, R.E. (1979) 'Kinetics of Breakpoint Chlorination in Continuous Flow Systems.', *J Am Water Works Assoc*, 71(3), pp. 164–172. Available at: <https://doi.org/10.1002/j.1551-8833.1979.tb04322.x>.

scikit-learn developers (2023) 'scikit-learn: Machine Learning in Python'. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>.

Sharma, P. (2022) 'Random forest prediction scheme'.

Shorten, C. and Khoshgoftaar, T.M. (2019) 'A survey on Image Data Augmentation for Deep Learning', *Journal of Big Data*, 6(1). Available at: <https://doi.org/10.1186/s40537-019-0197-0>.

Somanna (2023) 'Guide to adding noise to your data using Python and

NumPy'. Available at: https://medium.com/@ms_somanna/guide-to-adding-noise-to-your-data-using-python-and-numpy-c8be815df524.

Soyupak, S. *et al.* (2011) 'On the usage of artificial neural networks in chlorine control applications for water distribution networks with high quality water', *Journal of Water Supply: Research and Technology - AQUA*, 60(1), pp. 51–60. Available at: <https://doi.org/10.2166/aqua.2011.086>.

Specht, D.F. (1991) 'A general regression neural network', *IEEE transactions on neural networks*, 2(6), pp. 568–576.

Sun, H.J. *et al.* (2019) 'The toxicity of 2,6-dichlorobenzoquinone on the early life stage of zebrafish: A survey on the endpoints at developmental toxicity, oxidative stress, genotoxicity and cytotoxicity', *Environmental Pollution*, 245, pp. 719–724. Available at: <https://doi.org/10.1016/j.envpol.2018.11.051>.

Susanto, A. *et al.* (2020) 'Analysis of free residual chlorine in drinking water distribution systems in ore processing industry', *Polish Journal of Environmental Studies*, 29(6), pp. 4321–4330. Available at: <https://doi.org/10.15244/pjoes/120154>.

Swamynathan, M. (2019) *Mastering Machine Learning with Python in Six Steps*, *Mastering Machine Learning with Python in Six Steps*. Available at: <https://doi.org/10.1007/978-1-4842-4947-5>.

Turing (2022) *Need Deep Neural Networks Diagram*, *Turing Knowledge Base*. Available at: https://images.prismic.io/turing/659d78a0531ac2845a2742df_need_deep_neural_networks_5_11zon_54d2f93a48.webp?auto=format,compress.

Vaswani, A. *et al.* (2017) 'Attention is all you need', *Advances in Neural Information Processing Systems*, 2017-Decem(Nips), pp. 5999–6009.

Viera Valencia, L.F. and Garcia Giraldo, D. (2019) 済無No Title No Title No Title, *Angewandte Chemie International Edition*, 6(11), 951–952.

Wadkar, D. V. and Kote, A.S. (2017) 'Prediction of residual chlorine in a water treatment plant using generalized regression neural network', *International Journal of Civil Engineering and Technology*, 8(8), pp. 1264–1270.

Wadkar, D.V., Nangare, P. and Wagh, M.P. (2021) 'Evaluation of water treatment plant using Artificial Neural Network (ANN) case study of Pimpri Chinchwad Municipal Corporation (PCMC)', *Sustainable Water Resources Management*, 7(4), pp. 1–14. Available at: <https://doi.org/10.1007/s40899-021-00532-w>.

Wang, Y. *et al.* (2019) 'A hybrid ensemble method for pulsar candidate classification', *Astrophysics and Space Science*, 364(8). Available at: <https://doi.org/10.1007/s10509-019-3602-4>.

WHO (2017) *Principles and Practices of Drinking-water Chlorination*. Available at: http://www.searo.who.int/entity/water_sanitation/documents/chlorination_guide_presentation.pdf?ua=1.

World Health Organization (2017) *WHO report on the global tobacco epidemic, 2017: monitoring tobacco use and prevention policies*, World Health Organization. Available at: <https://apps.who.int/iris/bitstream/handle/10665/255874/9789241512824-eng.pdf>.

Wu, N. *et al.* (2020) 'Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case'. Available at: <http://arxiv.org/abs/2001.08317>.

Zarafshan Saman; Roozbahani, Abbas; Hashemy, Seyed Mehdi; Zarafshan, Payam; Etezadi, Hamed, P.J. (2021) 'Artificial Intelligence Hybrid Deep Learning Model for Groundwater Level Prediction Using MLP-ADAM.', *arXiv: Learning*, NA(NA), p. NA-NA.

Zhou, R. *et al.* (2019) 'Prediction Model of Credit default based on XGBoost algorithm', *Bond*, 0(10), p. 13. Available at: <https://doi.org/10.54254/2755-2721/96/20241285>.

Zhu, X. *et al.* (2025) 'Transformer-based approaches for neuroimaging: an in-depth review of their role in classification and regression tasks', *Reviews in the Neurosciences*, 36(2), pp. 209–228.

Appendices

Appendix A: Programming Codes

```
In [1]: from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import pandas as pd
import numpy as np

from sklearn.model_selection import GridSearchCV
from sklearn.datasets import make_regression
from sklearn.preprocessing import StandardScaler

#
df = pd.read_excel('D1_AUG_2.xlsx')
df
```

```
In [2]: #
df['date'] = pd.to_datetime(df['date'])
df['year'] = df['date'].dt.year
df['month'] = df['date'].dt.month
df['day'] = df['date'].dt.day

df['Time'] = pd.to_datetime(df['Time'], format='%H:%M:%S').dt.time

print(df)
from sklearn.preprocessing import LabelEncoder
#
label_encoder = LabelEncoder()

df['Point'] = label_encoder.fit_transform(df['Point'])
print(df['Point'].head())
```

S1 # Run in Scenario one

```
In [3]: X = df[['time', 'Dis', 'Turb', 'TDS', 'PH', 'EC', 'So4', 'month', 'year', 'day', 'Alk',
y = df['RC']
```

S2 # Run in Scenario two

```
In [4]: #X = df[['time', 'Dis', 'Turb', 'EC', 'TDS', 'Temp', 'month', 'year']]
#y = df['RC']
```

S3 # Run in Scenario three

```
In [5]: #X = df[['time', 'Dis', 'PH', 'So4', 'month', 'year', 'day', 'Alk', 'Cl', 'Ca', 'TH']]
#y = df['RC']
```

```

In [6]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_st

In [7]: scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

In [8]: # define RandomForestRegressor model
model = RandomForestRegressor()

# GridSearchCV
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

In [9]: rf = RandomForestRegressor(random_state=42)

In [10]: # GridSearchCV
grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=5, n_jobs=-1,

In [11]: # Conduct research
grid_result = grid_search.fit(X_train, y_train)

In [12]: #Print best result
print(f"Best: {grid_result.best_score_} using {grid_result.best_params_}")

Best: -0.45464103867372857 using {'max_depth': 20, 'min_samples_leaf': 1, 'min_sampl
es_split': 2, 'n_estimators': 200}

In [14]: # Create a new model using the best parameters.
best_model = RandomForestRegressor(n_estimators=200, max_depth=20, min_samples_spli

# Train the model using training data
best_model.fit(X_train, y_train)

# Evaluate the model on test data
y_pred = best_model.predict(X_test)

In [15]: from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np

# R2
r2 = r2_score(y_test, y_pred)

# MAE
mae = mean_absolute_error(y_test, y_pred)

# RMSE

```

```

rmse = np.sqrt(mean_squared_error(y_test, y_pred))

# MSE
mse = mean_squared_error(y_test, y_pred)

#
print(f'R²: {r2}')
print(f'Mean Absolute Error (MAE): {mae}')
print(f'Root Mean Squared Error (RMSE): {rmse}')
print(f'Mean Squared Error (MSE): {mse}')

```

```

R²: 0.9197143369191149
Mean Absolute Error (MAE): 0.3331157789204524
Root Mean Squared Error (RMSE): 0.4534557982436171
Mean Squared Error (MSE): 0.20562216096075595

```

```

In [16]: # Show results for some predictions
comparison = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
print(comparison.head())

```

	Actual	Predicted
255	2.335119	2.152411
114	3.700000	3.552029
314	3.205940	2.010465
268	4.751339	4.035268
167	1.363665	1.761611

```

In [17]: import matplotlib.pyplot as plt
# Plot actual values vs predicted values
plt.figure(figsize=(10, 6))
plt.plot(y_test.values, label='Actual Values', color='blue')
plt.plot(y_pred, label='Predicted Values', color='red', linestyle='--')

plt.title('Comparison of Actual and Predicted Values for RC using Random Forest')
plt.xlabel('Index')
plt.ylabel('RC')
plt.legend()

# Show the plot
plt.show()

```

For XGBoost Model

D1_AUG_2

```
In [1]: import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score
from scipy.stats import norm
import matplotlib.pyplot as plt

import xgboost as xgb
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.datasets import fetch_california_housing

from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

data = pd.read_excel('D1_AUG_2.xlsx')

data
```

```
In [2]: data['Point'] = pd.to_numeric(data['Point'], errors='coerce')

#
data['date'] = pd.to_datetime(data['date'])
data['year'] = data['date'].dt.year
data['month'] = data['date'].dt.month
data['day'] = data['date'].dt.day

# Converting time to minutes since the beginning of the day
```

```
In [3]: data['Time'] = pd.to_datetime(data['Time'], format='%H:%M:%S').dt.time

data['time'] = data['Time'].apply(lambda x: x.hour * 60 + x.minute)

print(data)
```

S1 # Run in Scenario one

```
In [4]: #Dividing the data into inputs and outputs
X = data[['time', 'Dis', 'PH', 'Turb', 'Temp', 'TDS', 'TH', 'Cl', 'So4', 'Alk', 'TDS', 'EC'],
y = data['RC']
```

S2 # Run in Scenario two

```
In [5]: # Dividing the data into inputs and outputs
#X = data[['time', 'Dis', 'Turb', 'Temp', 'TDS', 'EC', 'year', 'month', 'day']]
#y = data['RC']
```

S3 # Run in Scenario three

```
In [6]: # Dividing the data into inputs and outputs
#X = data[['time', 'Dis', 'PH', 'TH', 'Cl', 'So4', 'Alk', 'Ca', 'year', 'month', 'day']
#y = data['RC']
```

```
In [7]: # Dividing the data into a training set and a test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Standard scaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
In [8]: # Developing the XGBoost model (Regression)
xgb_regressor = xgb.XGBRegressor(objective="reg:squarederror", random_state=42)
```

```
In [9]: # Hyperparameter Optimization
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [3, 6, 9],
    'learning_rate': [0.01, 0.1, 0.2],
    'subsample': [0.8, 1.0],
    'colsample_bytree': [0.8, 1.0]
}
```

```
In [10]: # GridSearchCV
grid_search = GridSearchCV(
    estimator=xgb_regressor,
    param_grid=param_grid,
    scoring='neg_mean_squared_error',
    cv=5,
    verbose=1,
    n_jobs=-1
)

grid_search.fit(X_train, y_train)

# Print the Best Parameters
print("Best Parameters:", grid_search.best_params_)
```

Fitting 5 folds for each of 108 candidates, totalling 540 fits

Best Parameters: {'colsample_bytree': 0.8, 'learning_rate': 0.1, 'max_depth': 9, 'n_estimators': 200, 'subsample': 0.8}

```
In [11]: best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)
```

```
In [12]: # Performance metric
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f" ♦ MSE: {mse:.4f}")
print(f" ♦ RMSE: {rmse:.4f}")
print(f" ♦ MAE: {mae:.4f}")
print(f" ♦ R2 Score: {r2:.4f}")
```

- ◆ MSE: 0.1035
- ◆ RMSE: 0.3217
- ◆ MAE: 0.1998
- ◆ R² Score: 0.9596

```
In [13]: import matplotlib.pyplot as plt
# Plot actual values vs predicted values
plt.figure(figsize=(10, 6))
plt.plot(y_test.values, label='Actual Values', color='blue')
plt.plot(y_pred, label='Predicted Values', color='RED', linestyle='--')

plt.title('Comparison of Actual and Predicted Values for RC using XGBoost')
plt.xlabel('Index')
plt.ylabel('RC')
plt.legend()

# Show the plot
plt.show()
```

For GRNN Model

GRNN D1 D1_AUG_2

```
In [1]: # Import Libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from scipy.stats import norm
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

```
In [2]: df = pd.read_excel('D1_AUG_2.xlsx')

# Display the first few rows of the dataset
print("First 5 rows of the dataset:")
print(df.head())
```

```
In [3]: df['date'] = pd.to_datetime(df['date'])
df['year'] = df['date'].dt.year
df['month'] = df['date'].dt.month
df['day'] = df['date'].dt.day

df['Time'] = pd.to_datetime(df['Time'], format='%H:%M:%S').dt.time

#
df['time'] = df['Time'].apply(lambda x: x.hour * 60 + x.minute)

#
print(df)
```

```

In [4]: #S1      # Run in Scenario one

In [5]: # Divide data into features and purpose
X = df[['time', 'Dis', 'PH', 'Turb', 'Temp', 'TH', 'Cl', 'So4', 'Alk', 'TDS', 'EC', 'Ca', 'yea
y = df['RC']

In [6]: #S2      # Run in Scenario two

In [7]: #X = df[['time', 'Dis', 'Turb', 'EC', 'TDS', 'Temp', 'month', 'year', 'day']]
#y = df['RC']

In [8]: #S3      # Run in Scenario three

In [9]: #X = df[['time', 'Dis', 'PH', 'So4', 'month', 'year', 'day', 'Alk', 'CL', 'Ca', 'TH']]
#y = df['RC']

In [10]: # Splitting data into training and testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_sta

# (Standardization)
scaler_standard = StandardScaler()
X_train_scaled = scaler_standard.fit_transform(X_train)
X_test_scaled = scaler_standard.transform(X_test)

In [11]: # GRNN function for prediction
class GRNN:
    def __init__(self, sigma):
        self.sigma = sigma

    def fit(self, X, y):
        self.X_train = X
        self.y_train = y

    def predict(self, X):
        y_pred = []
        for x in X:
            # Calculate the distance between the sample and the stored points
            distances = np.linalg.norm(self.X_train - x, axis=1)
            weights = np.exp(-distances**2 / (2 * self.sigma**2))
            weighted_sum = np.sum(weights * self.y_train)
            total_weight = np.sum(weights)
            y_pred.append(weighted_sum / total_weight)
        return np.array(y_pred)

#Build the model
sigma = 0.2
model = GRNN(sigma=sigma)
model.fit(X_train_scaled, y_train)

```

```
# Prediction on test data
y_pred = model.predict(X_test_scaled)
```

```
In [12]: # Calculate MSE and R^2
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error (MSE): {mse:.4f}")
print(f"R-squared (R^2): {r2:.4f}")
```

Mean Squared Error (MSE): 0.0097
R-squared (R^2): 0.9962

```
In [13]: #MAE account
mae = mean_absolute_error(y_test, y_pred)

# RMSE calculation
rmse = np.sqrt(mse)

# Print results
print(f"Mean Absolute Error (MAE): {mae:.4f}")

print(f"Root Mean Squared Error (RMSE): {rmse:.4f}")
```

Mean Absolute Error (MAE): 0.0683
Root Mean Squared Error (RMSE): 0.0982

```
In [14]: import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.plot(y_test.values, label='Actual Values', color='blue', linestyle='-')
plt.plot(y_pred, label='Predicted Values', color='red', linestyle='--')
plt.title('Actual vs Predicted Values for RC using GRNN')
plt.xlabel('Sample Index')
plt.ylabel('RC Value')
plt.legend()
plt.show()
```

For MLP Model

DI_AUG_2

```
In [1]: import numpy as np
import tensorflow as tf
import random

np.random.seed(42)
tf.random.set_seed(42)
random.seed(42)
```

```
In [2]: import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
#
data = pd.read_excel('D1_AUG_2.xlsx')

data
```

```
In [3]: #Convert date to year month day
data['date'] = pd.to_datetime(data['date'])
data['year'] = data['date'].dt.year
data['month'] = data['date'].dt.month
data['day'] = data['date'].dt.day

data['Time'] = pd.to_datetime(data['Time'], format='%H:%M:%S').dt.time
data['time'] = data['Time'].apply(lambda x: x.hour * 60 + x.minute)
print(data)
```

S1

Run in Scenario one

```
In [4]: X = data[['time', 'Dis', 'PH', 'Turb', 'Temp', 'TH', 'Cl', 'So4', 'Alk', 'TDS', 'EC', 'Ca']]
y = data['RC']
```

S2 **# Run in Scenario two**

```
In [5]: #X = data[['time', 'Dis', 'Turb', 'EC', 'TDS', 'Temp', 'month', 'year', 'day']]
#y = data['RC']
```

S3 **# Run in Scenario three**

```
In [6]: #X = data[['time', 'Dis', 'PH', 'So4', 'month', 'year', 'day', 'Alk', 'Cl', 'Ca', 'TH']]
#y = data['RC']
```

```
In [7]: #divided data in to training and testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_sta

# StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
In [8]: from sklearn.neural_network import MLPRegressor
        from sklearn.model_selection import GridSearchCV

        # define the Model with early_stopping
        mlp = MLPRegressor(max_iter=5000, early_stopping=True, n_iter_no_change=20)

        # Define the values to be tested in GridSearchCV
        param_grid = {
            'hidden_layer_sizes': [(50, 50), (100, 50), (100, 100), (128, 64, 32)],
            'activation': ['relu', 'tanh'],
            'alpha': [0.0001, 0.001, 0.01],
            'solver': ['adam']
        }

        # GridSearchCV
        grid_search = GridSearchCV(mlp, param_grid, cv=3, scoring='r2', n_jobs=-1)
        grid_search.fit(X_train_scaled, y_train)
        #
        print(f"Best hyperparameters: {grid_search.best_params_}")

Best hyperparameters: {'activation': 'relu', 'alpha': 0.001, 'hidden_layer_sizes':
(100, 50), 'solver': 'adam'}
```

```
In [9]: # Use the best evaluation model
        best_mlp = grid_search.best_estimator_
        score = best_mlp.score(X_test_scaled, y_test)
        print(f"R2 Score on Test Set: {score:.4f}")
```

R² Score on Test Set: 0.7304

```
In [10]: # Use the best evaluation model
        best_mlp = grid_search.best_estimator_
        score = best_mlp.score(X_test_scaled, y_test)
        print(f"R2 Score on Test Set: {score:.4f}")
        best_mlp = grid_search.best_estimator_
        y_pred = best_mlp.predict(X_test_scaled)
```

```
# MSE, MAE, RMSE, R2
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f"MSE: {mse:.4f}")
print(f"MAE: {mae:.4f}")
print(f"RMSE: {rmse:.4f}")
print(f"R2 Score: {r2:.4f}")
```

R² Score on Test Set: 0.7304
MSE: 0.7949
MAE: 0.4809
RMSE: 0.8916
R² Score: 0.7304

```
In [13]: import matplotlib.pyplot as plt
# Plot actual values vs predicted values
plt.figure(figsize=(10, 6))
plt.plot(y_test.values, label='Actual Values', color='blue')
plt.plot(y_pred, label='Predicted Values', color='RED', linestyle='--')

plt.title('Comparison of Actual and Predicted Values for RC using MLP')
plt.xlabel('Index')
plt.ylabel('RC')
plt.legend()

# Show the plot
plt.show()
```

Transformer model

```
In [1]: import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import torch
import torch.nn as nn
from torch.utils.data import TensorDataset, DataLoader
from skorch import NeuralNetRegressor

df = pd.read_excel('D1_AUG_2 Transformer.xlsx')

df['date'] = pd.to_datetime(df['date'])
#df = df.sort_values('date').reset_index(drop=True)
df['day'] = df['date'].dt.day
df['month'] = df['date'].dt.month
df['year'] = df['date'].dt.year
df
```

```
Out[1]:
```

date	Point	Time	RC	Dis	Turb	PH	Temp	TDS
------	-------	------	----	-----	------	----	------	-----

```
In [2]: df['Time'] = pd.to_datetime(df['Time'], format='%H:%M:%S').dt.time
df['time'] = df['Time'].apply(lambda x: x.hour * 60 + x.minute)
print(df)
```

```
In [3]: df = df[~df['date'].isin([pd.Timestamp('2090-01-01'), pd.Timestamp('2090-01-02')])]
```

S1 # Run in Scenario one

```
In [4]: #target = 'RC'
#features = ['Dis', 'Turb', 'PH', 'Temp', 'TDS', 'EC', 'TH', 'Ca', 'Cl', 'So4', 'AL
```

S2 # Run in Scenario two

```
In [5]: #target = 'RC'
#features = ['Dis', 'Turb', 'Temp', 'TDS', 'EC', 'month', 'year', 'time', 'day']
```

S3 # Run in Scenario Three

```
In [6]: target = 'RC'
features = ['Dis', 'PH', 'TH', 'Ca', 'Cl', 'So4', 'Alk', 'month', 'year', 'time', 'day']
```

```
In [7]: scaler_X = StandardScaler()
scaler_y = StandardScaler()
X_scaled = scaler_X.fit_transform(df[features])
y_scaled = scaler_y.fit_transform(df[[target]])
```

```
In [8]: def create_sequences(X, y, window_size):
X_seq, y_seq = [], []
for i in range(len(X) - window_size):
X_seq.append(X[i:i+window_size])
y_seq.append(y[i+window_size])
return np.array(X_seq), np.array(y_seq)

window_size = 8
X_seq, y_seq = create_sequences(X_scaled, y_scaled, window_size)
```

```
# نموذج Transformer
class TimeSeriesTransformer(nn.Module):
    def __init__(self, input_dim, d_model, nhead, num_layers, dropout=0.1):
        super(TimeSeriesTransformer, self).__init__()
        self.input_projection = nn.Linear(input_dim, d_model)
        encoder_layer = nn.TransformerEncoderLayer(d_model=d_model, nhead=nhead, dr
        self.transformer_encoder = nn.TransformerEncoder(encoder_layer, num_layers=
        self.fc = nn.Linear(d_model, 1)

    def forward(self, x):
        x = self.input_projection(x)
        x = self.transformer_encoder(x)
        x = x.mean(dim=1)
        out = self.fc(x)
        return out
```

```

In [14]: # GridSearchCV
net = NeuralNetRegressor(
    module=TimeSeriesTransformer,
    module__input_dim=len(features),
    max_epochs=50,
    lr=0.001,
    optimizer=torch.optim.Adam,
    criterion=nn.MSELoss,
    batch_size=32,
    device='cuda' if torch.cuda.is_available() else 'cpu'
)

params = {
    'module__d_model': [64, 128],

    'module__nhead': [2,4],
    'module__num_layers': [2,4],
    'lr': [0.01,0.001,0.0001]
}

X_np = X_seq.astype(np.float32)
y_np = y_seq.astype(np.float32).reshape(-1, 1)

gs = GridSearchCV(net, params, refit=True, cv=3, scoring='neg_mean_squared_error',
gs.fit(X_np, y_np)

print("The best Hyperparameters:", gs.best_params_)
print("Min MSE:", -gs.best_score_)

```

The best Hyperparameters: {'lr': 0.001, 'module__d_model': 128, 'module__nhead': 2, 'module__num_layers': 2}
Min MSE: 0.1864981402953466

```

In [15]: # Evaluating
X_test_seq = X_test.astype(np.float32)
y_test_seq = y_test.astype(np.float32)

best_model = gs.best_estimator_
y_pred = best_model.predict(X_test_seq)

if y_test_seq.ndim > 2:
    y_test_seq = y_test_seq.reshape(y_test_seq.shape[0], -1)

mse = mean_squared_error(y_test_seq, y_pred)
mae = mean_absolute_error(y_test_seq, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test_seq, y_pred)

print(f"MSE: {mse:.4f}")
print(f"MAE: {mae:.4f}")
print(f"RMSE: {rmse:.4f}")
print(f"R² Score: {r2:.4f}")

MSE: 0.0723
MAE: 0.1867
RMSE: 0.2689
R² Score: 0.9183

```

Appendix B: Training Model Result

1. RF_D1_S1

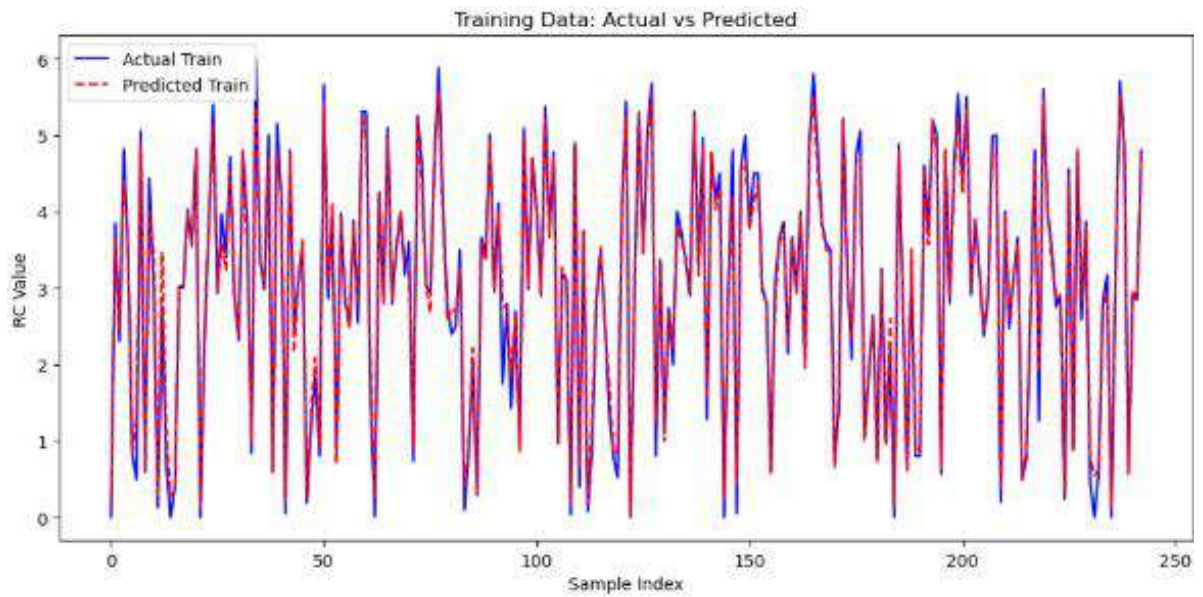


Figure (B.1): Comparison of Actual and Predicted Values for RF_D1_S1 /Training

Table (B.1): Performance metric for RF_D1_S1/Training

RF-D1-S1	Training Results
MAE	0.146
MSE	0.044
RMSE	0.211
R ²	0.983

2. RF_D1_S2

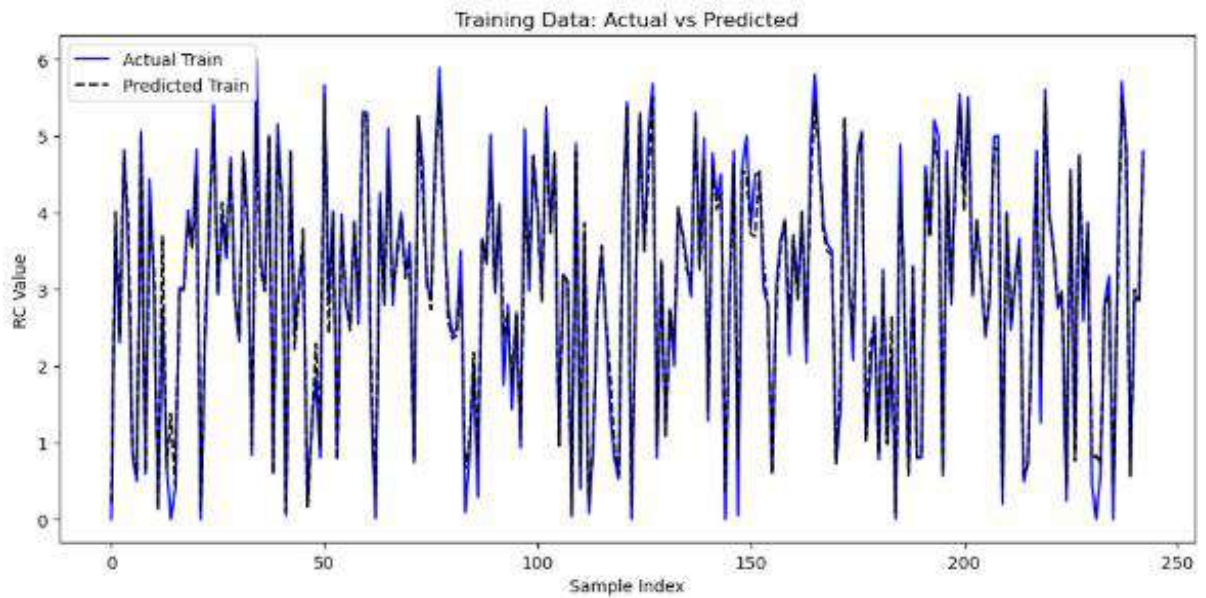


Figure (B.2): Comparison of Actual and Predicted Values for RF_D1_S2 /Training

Table (B.2): Performance metric for RF_D1_S2/Training

RF-D1-S2	Training Results
MAE	0.159
MSE	0.062
RMSE	0.250
R ²	0.977

3. RF_D1_S3

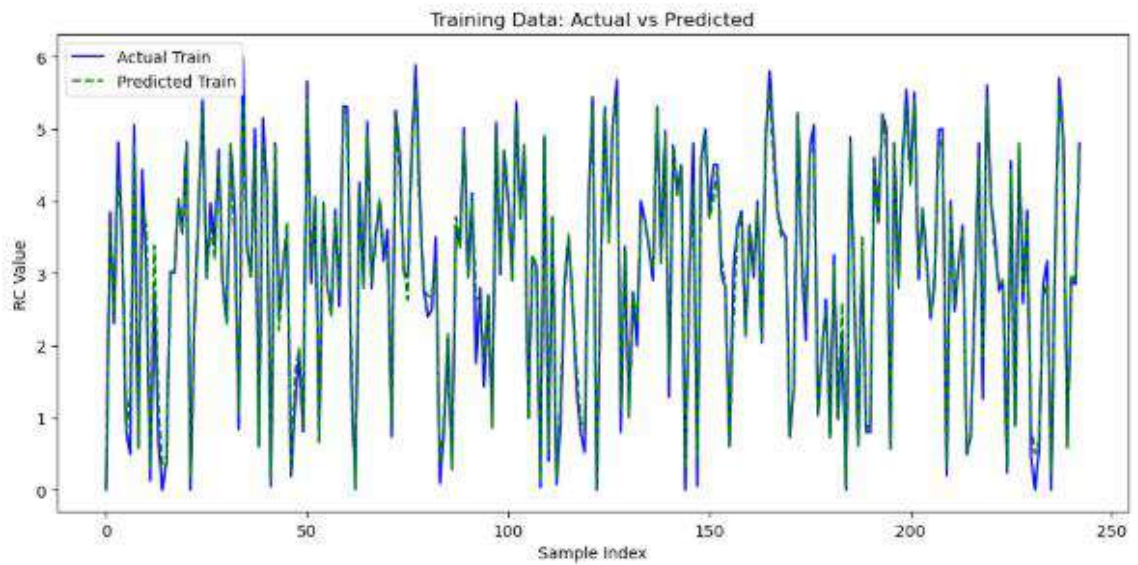


Figure (B. 3): Comparison of Actual and Predicted Values for RF_D1_S3 /Training

Table (B.3): Performance metric for RF_D1_S3/Training

RF-D1-S2	Training Results
MAE	0.144
MSE	0.044
RMSE	0.210
R ²	0.983

4. RF_D2_S1

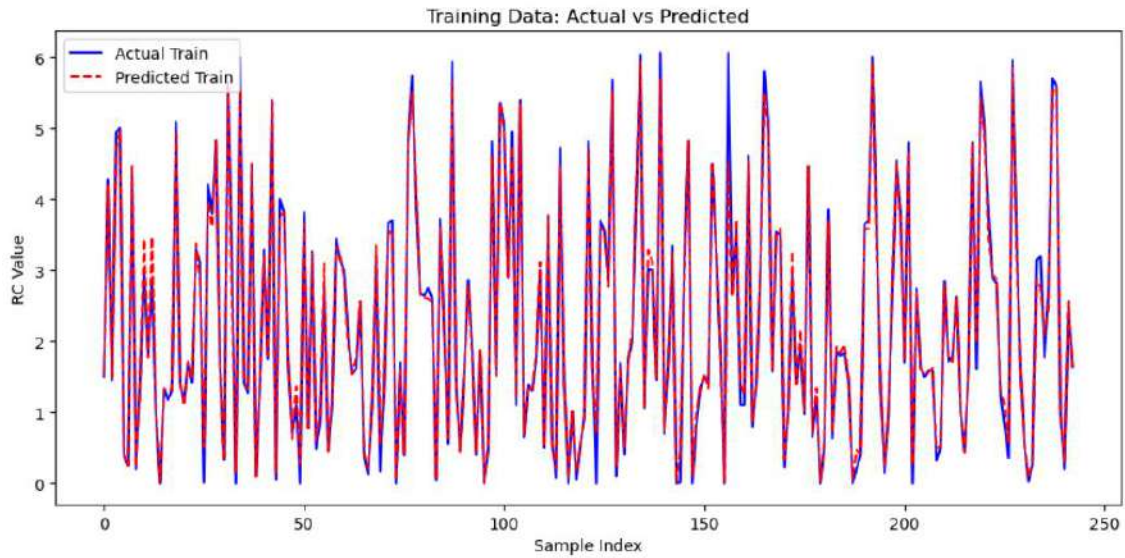


Figure (B.4): Comparison of Actual and Predicted Values for RF_D2_S1/Training

Table (B. 4): Performance metric for RF_D2_S1/Training

RF-D2-S1	Training Results
MAE	0.111
MSE	0.034
RMSE	0.184
R ²	0.988

5. RF_D2_S2

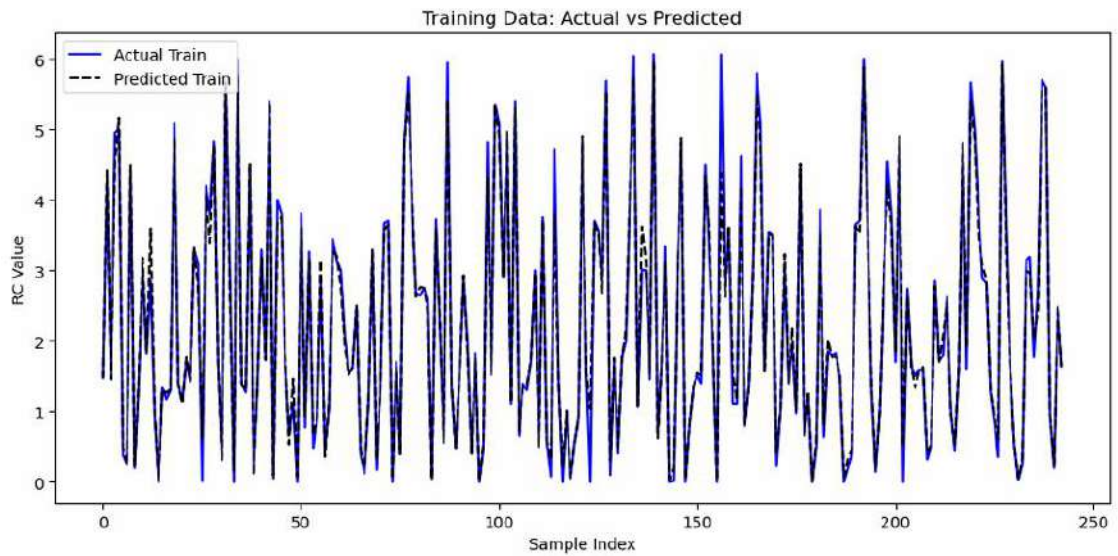


Figure (B.5): Comparison of Actual and Predicted Values for RF_D2_S2/Training

Table (B.5): Performance metric for RF_D2_S2/Training

RF-D2-S2	Training Results
MAE	0.124
MSE	0.046
RMSE	0.216
R^2	0.984

6. RF_D2_S3

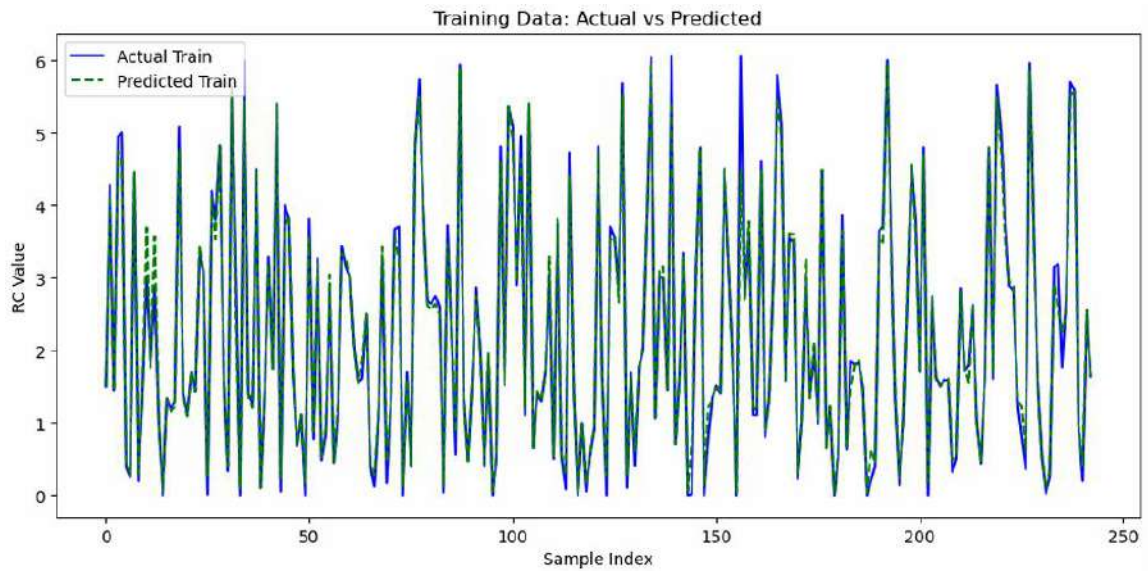


Figure (B.6): Comparison of Actual and Predicted Values for RF_D2_S3/Training

Table (B.6): Performance metric for RF_D2_S3/Training

RF-D2-S3	Training Results
MAE	0.122
MSE	0.042
RMSE	0.205
R ²	0.985

7. XGBoost_D1_S1

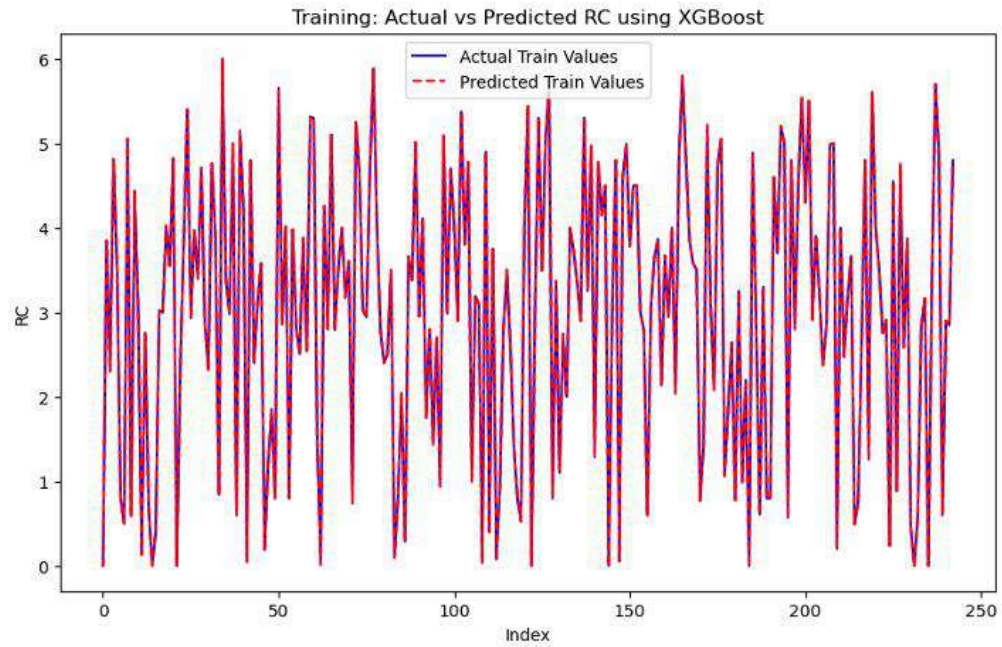


Figure (B.7): Comparison of Actual and Predicted Values for XGBoost_D1_S1 /Training

Table (B.7): Performance metric for XGBoost_D1_S1/Training

XGBoost-D1-S1	Training Results
MAE	0.0005
MSE	0.000
RMSE	0.0008
R ²	1.000

8. XGBoost_D1_S2

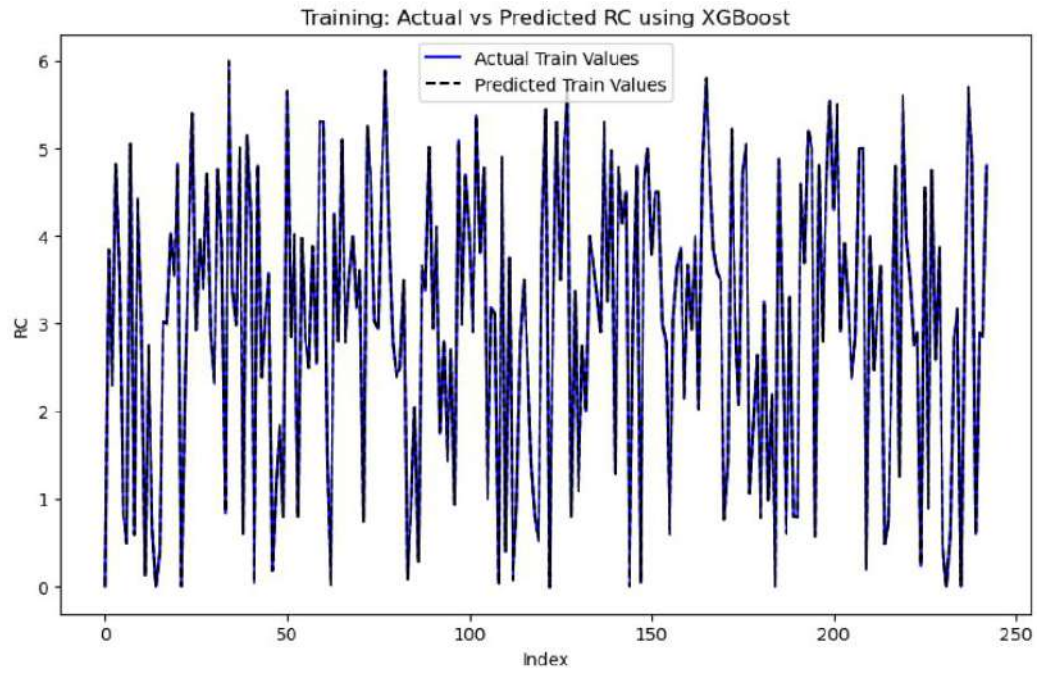


Figure (B. 8): Comparison of Actual and Predicted Values for XGBoost_D1_S2 /Training

Table (B. 8): Performance metric for XGBoost_D1_S2/Training

XGBoost-D1-S2	Training Results
MAE	0.005
MSE	0.000
RMSE	0.007
R²	1.000

9. XGBoost_D1_S3

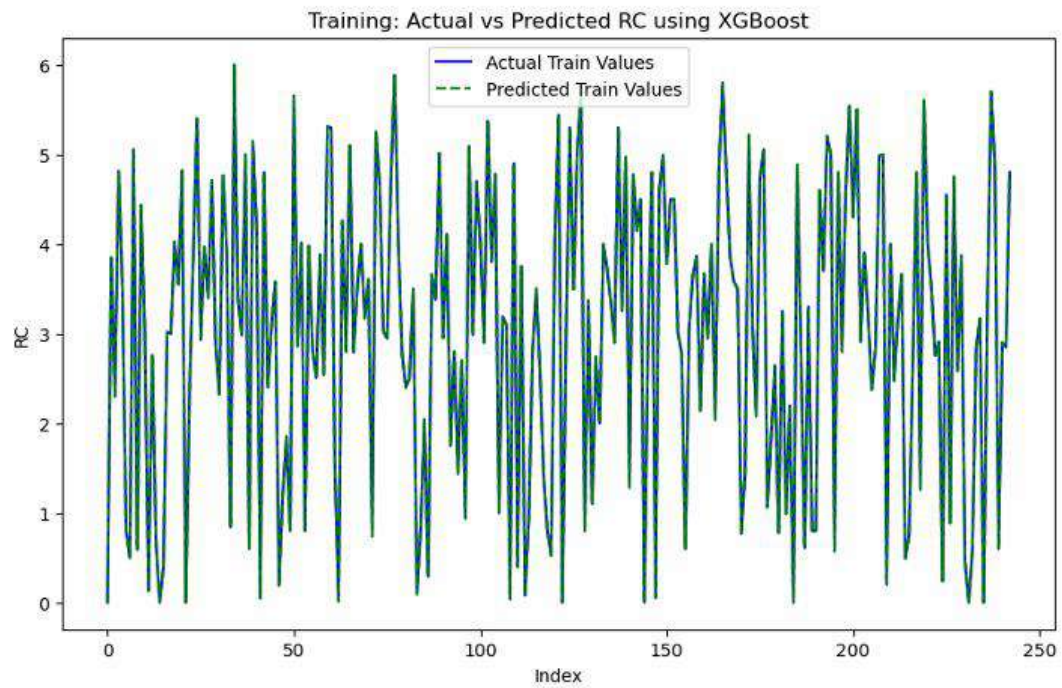


Figure (B.9): Comparison of Actual and Predicted Values for XGBoost_D1_S3 /Training

Table (B.9): Performance metric for XGBoost_D1_S3/Training

XGBoost-D1-S3	Training Results
MAE	0.0005
MSE	0.000
RMSE	0.0007
R ²	1.000

10. XGBoost_D2_S1

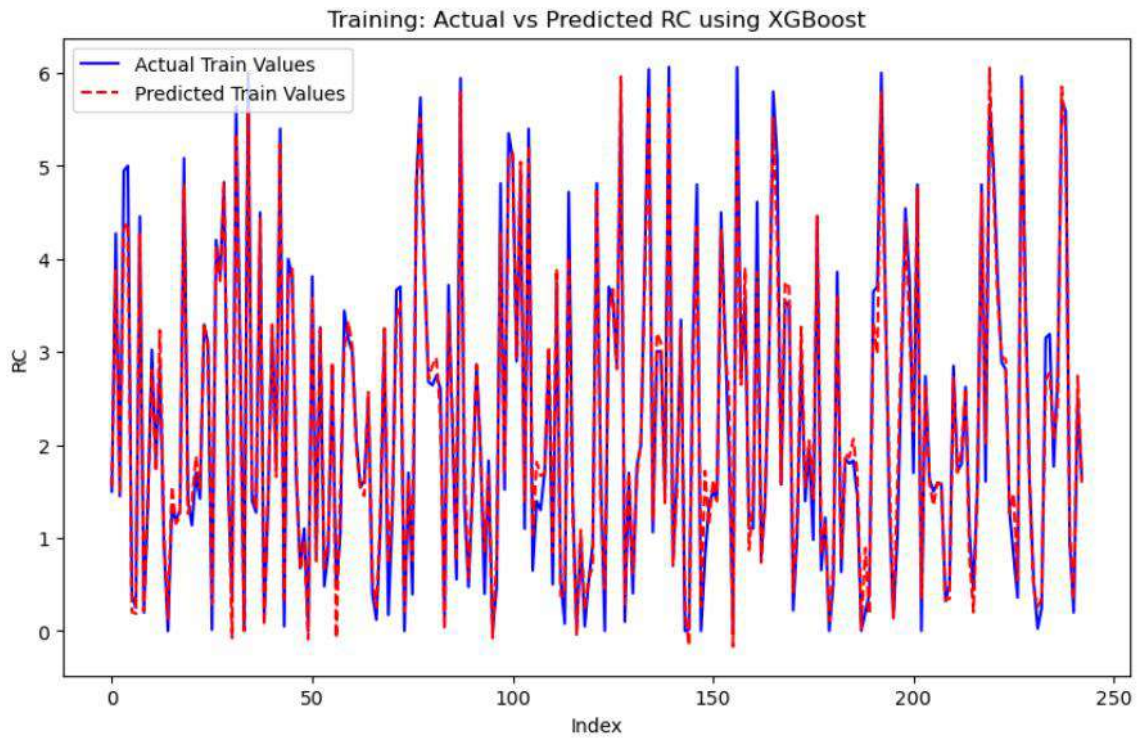


Figure (B.10): Comparison of Actual and Predicted Values for XGBoost_D2_S1/Training

Table (B.10): Performance metric for XGBoost_D2_S1/Training

XGBoost-D2-S1	Training Results
MAE	0.197
MSE	0.069
RMSE	0.262
R²	0.976

11. XGBoost_D2_S2

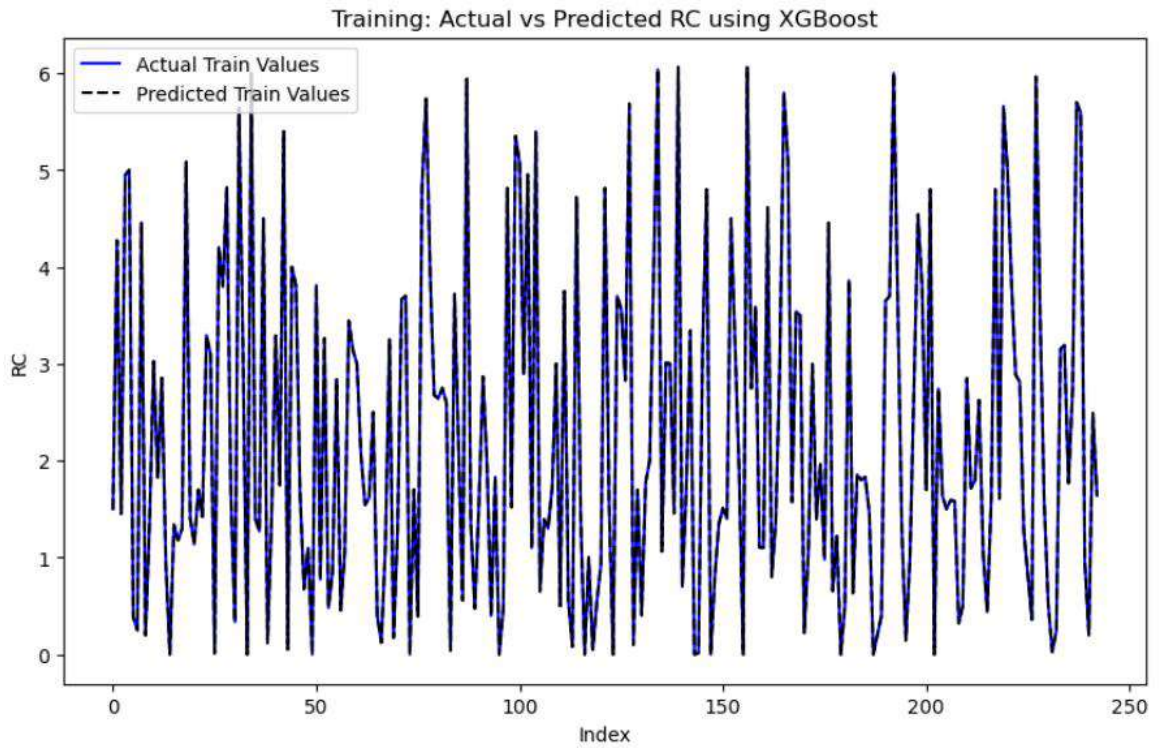


Figure (B. 11): Comparison of Actual and Predicted Values for XGBoost_D2_S2/Training

Table (B.11): Performance metric for XGBoost_D2_S2/Training

XGBoost-D2-S2	Training Results
MAE	0.0007
MSE	0.000
RMSE	0.0009
R^2	1.000

12. XGBoost_D2_S3

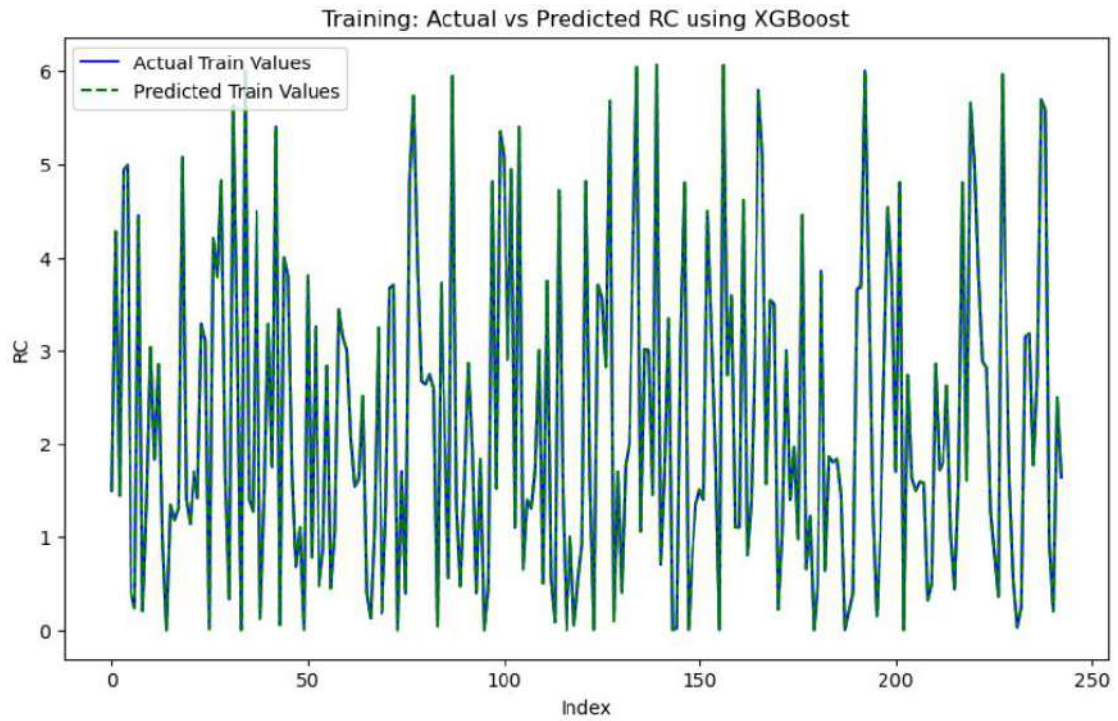


Figure (B.12): Comparison of Actual and Predicted Values for XGBoost_D2_S3/Training

Table (B.12): Performance metric for XGBoost_D2_S3/Training.

XGBoost-D2-S3	Training Results
MAE	0.004
MSE	0.000
RMSE	0.005
R²	1.000

13. MLP_D1_S1

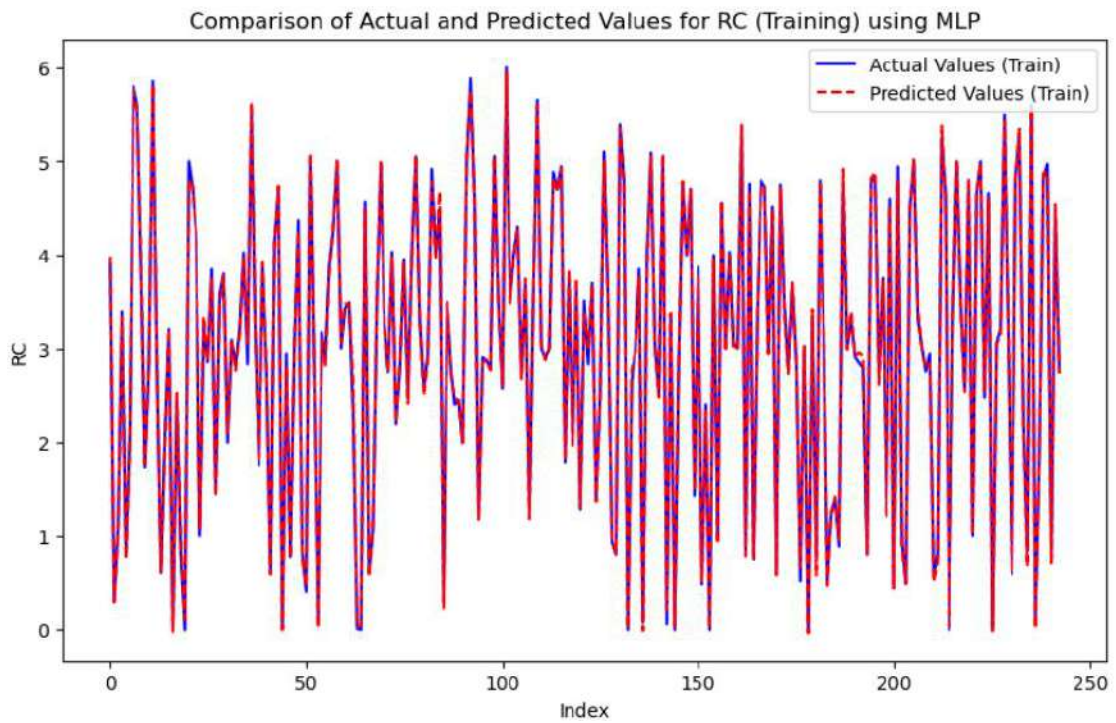


Figure (B.13): Comparison of Actual and Predicted Values for MLP_D1_S1/Training.

Table (B.13): Performance metric for MLP_D1_S1/Training.

MLP-D1-S1	Training Results
MAE	0.054
MSE	0.005
RMSE	0.072
R ²	0.997

14. MLP_D1_S2

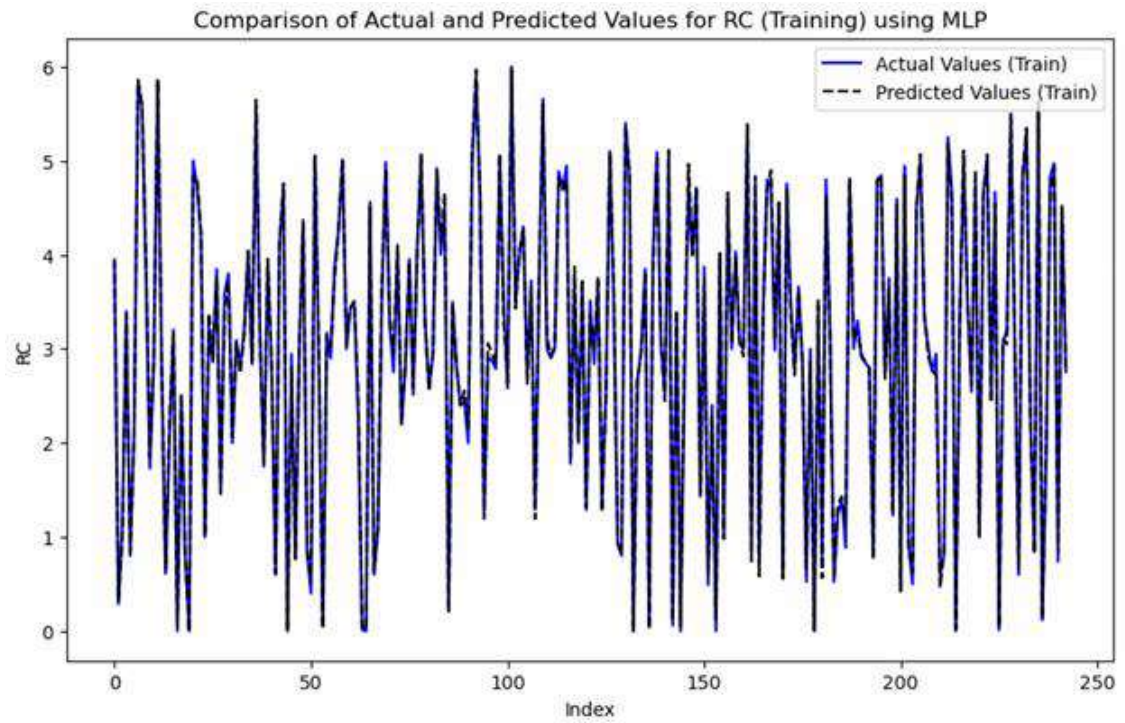


Figure (B. 14): Comparison of Actual and Predicted Values for MLP_D1_S2/Training.

Table (B.14): Performance metric for MLP_D1_S2/Training.

MLP-D1-S2	Training Results
MAE	0.066
MSE	0.007
RMSE	0.085
R^2	0.997

15. MLP_D1_S3

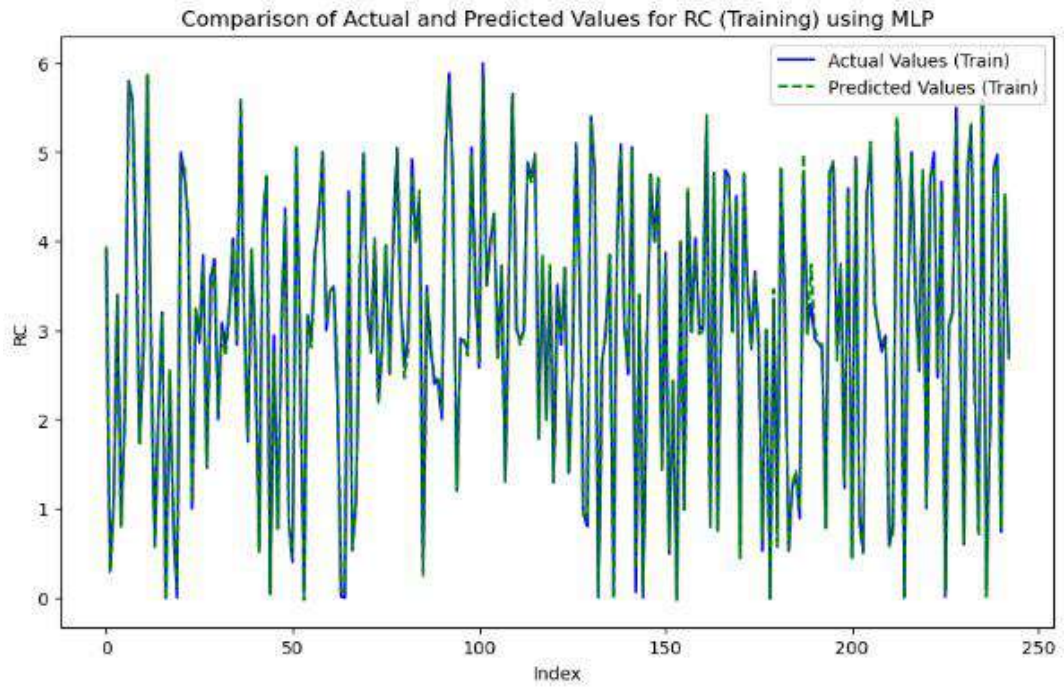


Figure (B. 15): Comparison of Actual and Predicted Values for MLP_D1_S3/Training.

Table (B.15): Performance metric for MLP_D1_S3/Training.

MLP-D1-S3	Training Results
MAE	0.057
MSE	0.006
RMSE	0.079
R ²	0.997

16. MLP_D2_S1

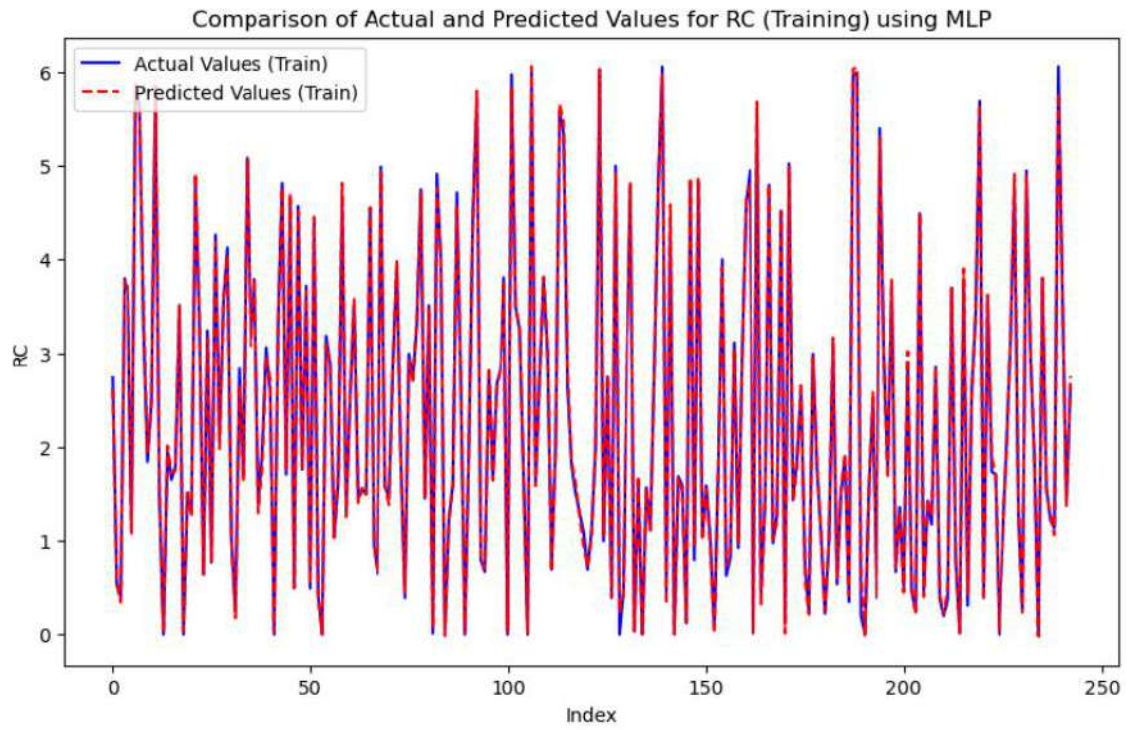


Figure (B.16): Comparison of Actual and Predicted Values for MLP_D2_S1/Training.

Table (B. 16): Performance metric for MLP_D2_S1/Training.

MLP -D2-S1	Training Results
MAE	0.057
MSE	0.006
RMSE	0.077
R^2	0.998

17. MLP_D2_S2

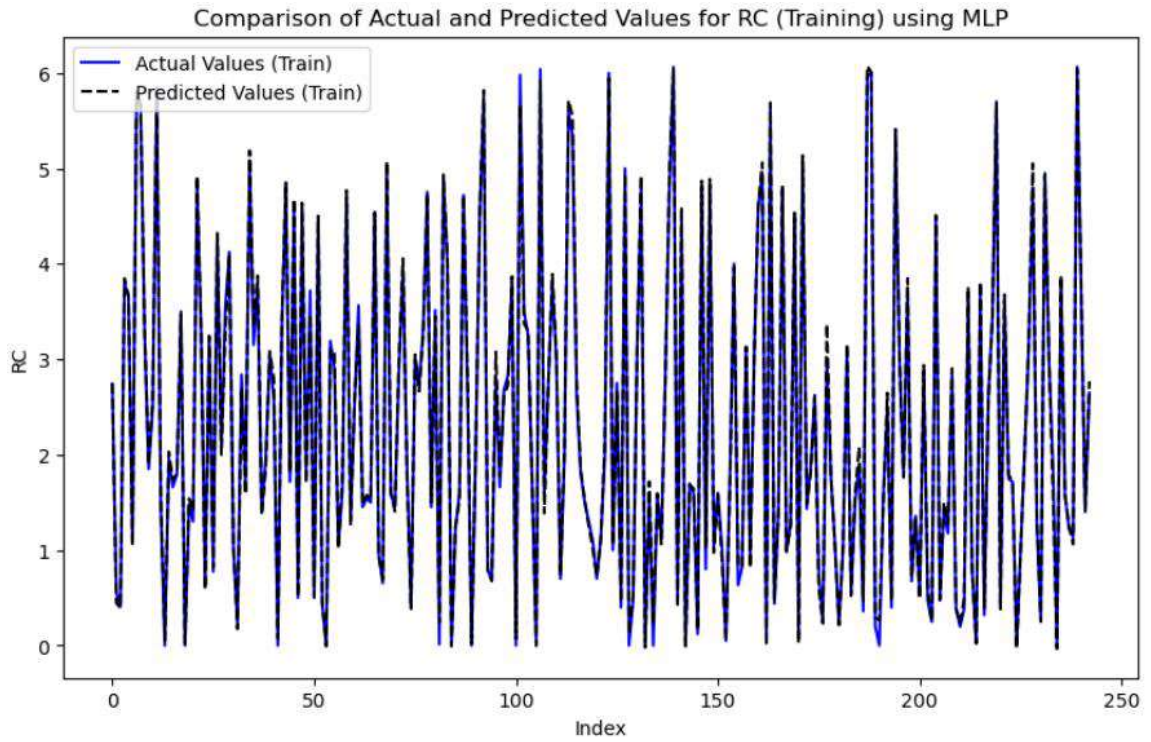


Figure (B. 17): Comparison of Actual and Predicted Values for MLP_D2_S2/Training.

Table (B. 17): Performance metric for MLP_D2_S2/Training.

MLP -D2-S2	Training Results
MAE	0.065
MSE	0.008
RMSE	0.091
R^2	0.997

18. MLP_D2_S3

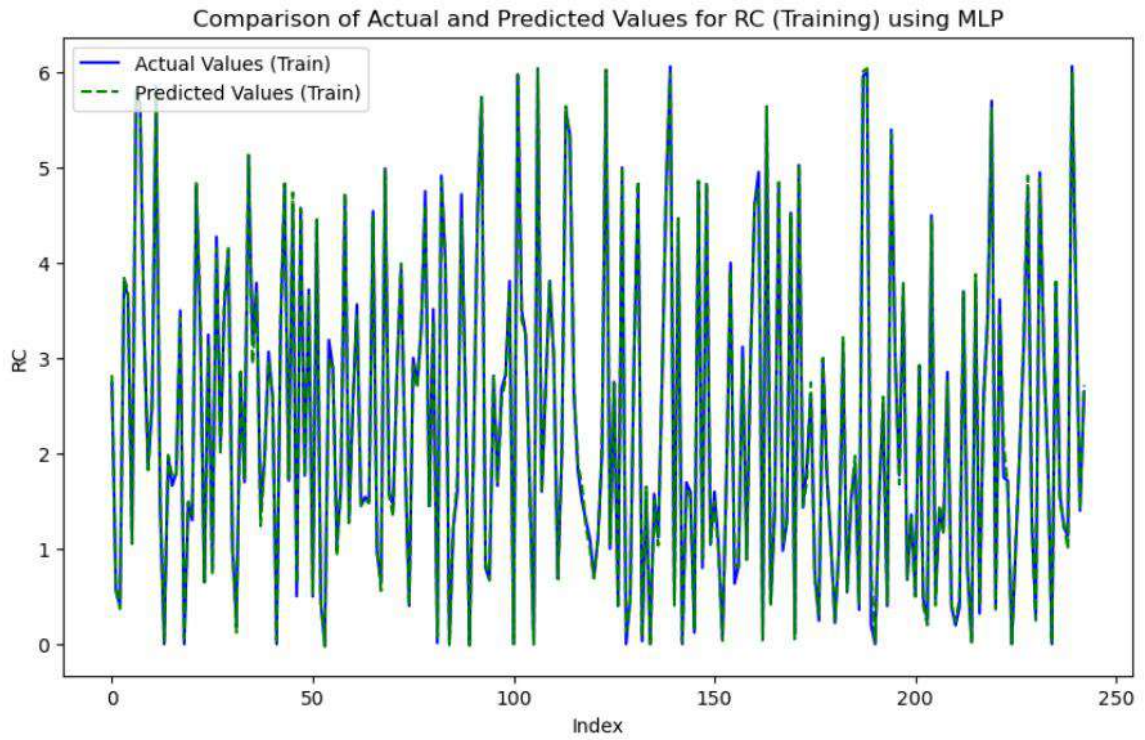


Figure (B. 18): Comparison of Actual and Predicted Values for MLP_D2_S3/Training.

Table (B. 18): Performance metric for MLP_D2_S3/Training.

MLP -D2-S3	Training Results
MAE	0.057
MSE	0.006
RMSE	0.081
R^2	0.997

19. GRNN D1_S1

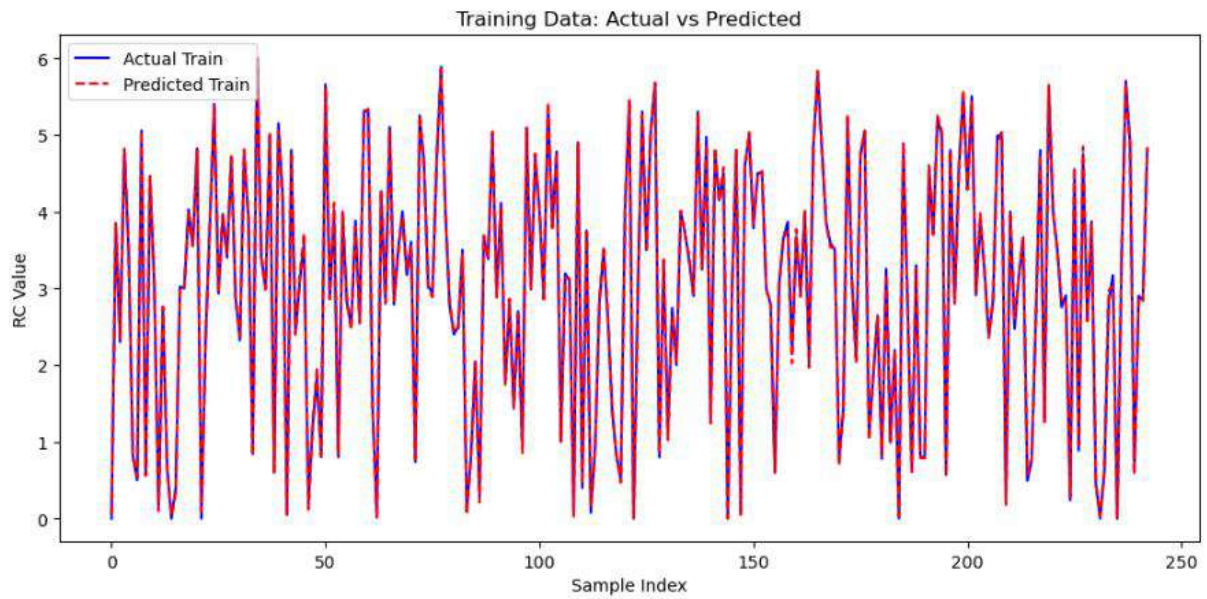


Figure (B.19): Comparison of Actual and Predicted Values for GRNN_D1_S1 /Training.

Table (B.19): Performance metric for GRNN_D1_S1/Training

GRNN-D1-S1	Training Results
MAE	0.036
MSE	0.002
RMSE	0.047
R^2	0.999

20. GRNN_D1_S2

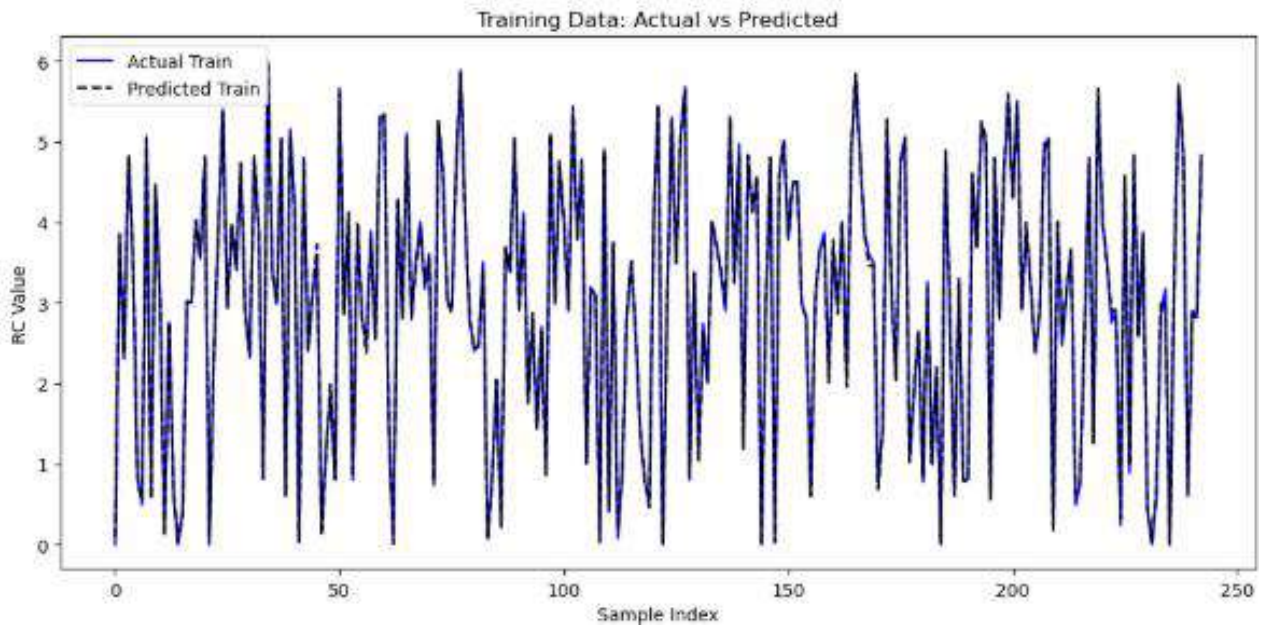


Figure (B.20): Comparison of Actual and Predicted Values for GRNN_D1_S2 /Training.

Table (B. 20): Performance metric for GRNN_D1_S2/Training

GRNN-D1-S2	Training Results
MAE	0.041
MSE	0.003
RMSE	0.054
R^2	0.998

21. GRNN-D1-S3

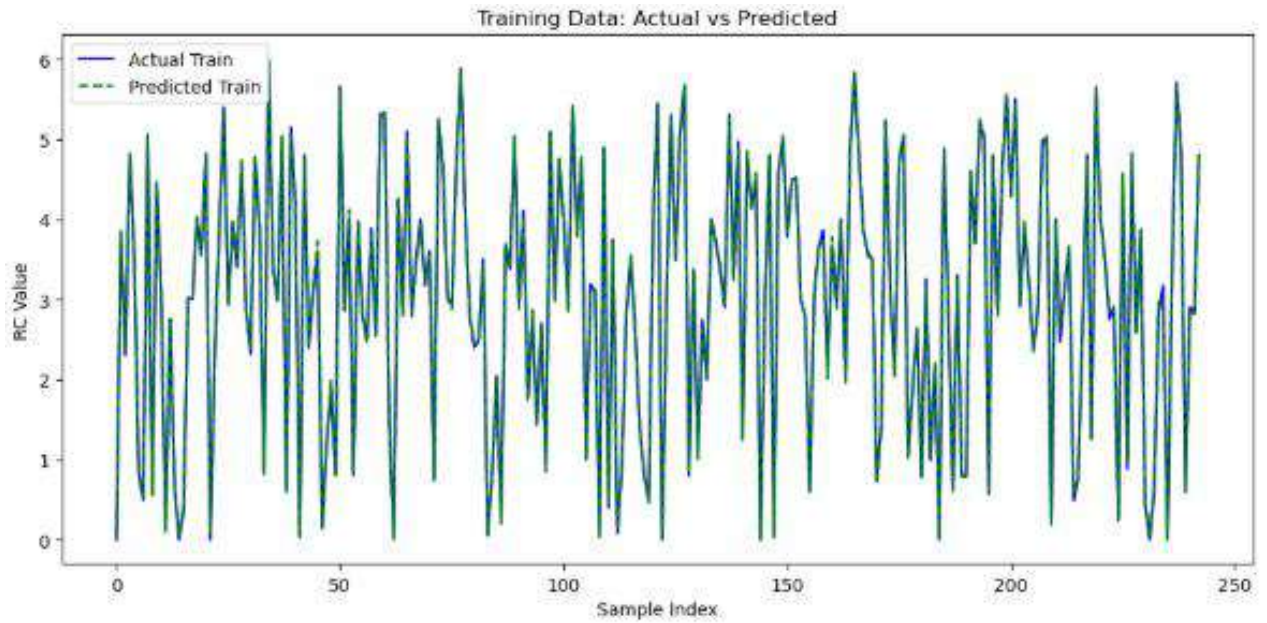


Figure (B.21): Comparison of Actual and Predicted Values for GRNN_D1_S3 /Training.

Table (B. 21): Performance metric for GRNN_D1_S3/Training

GRNN-D1-S3	Training Results
MAE	0.039
MSE	0.002
RMSE	0.051
R ²	0.999

22. GRNN_D2_S1

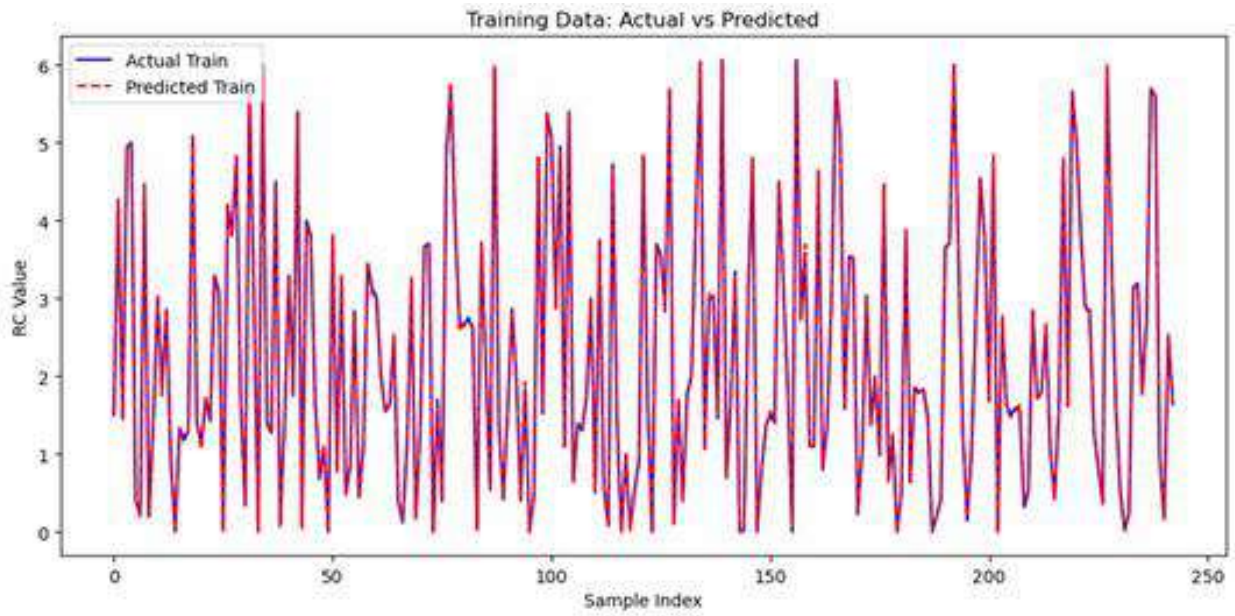


Figure (B.22): Comparison of Actual and Predicted Values for GRNN_D2_S1/Training.

Table (B. 22): Performance metric for GRNN_D2_S1/Training

GRNN-D2-S1	Training Results
MAE	0.026
MSE	0.001
RMSE	0.035
R ²	0.999

23. GRNN_D2_S2

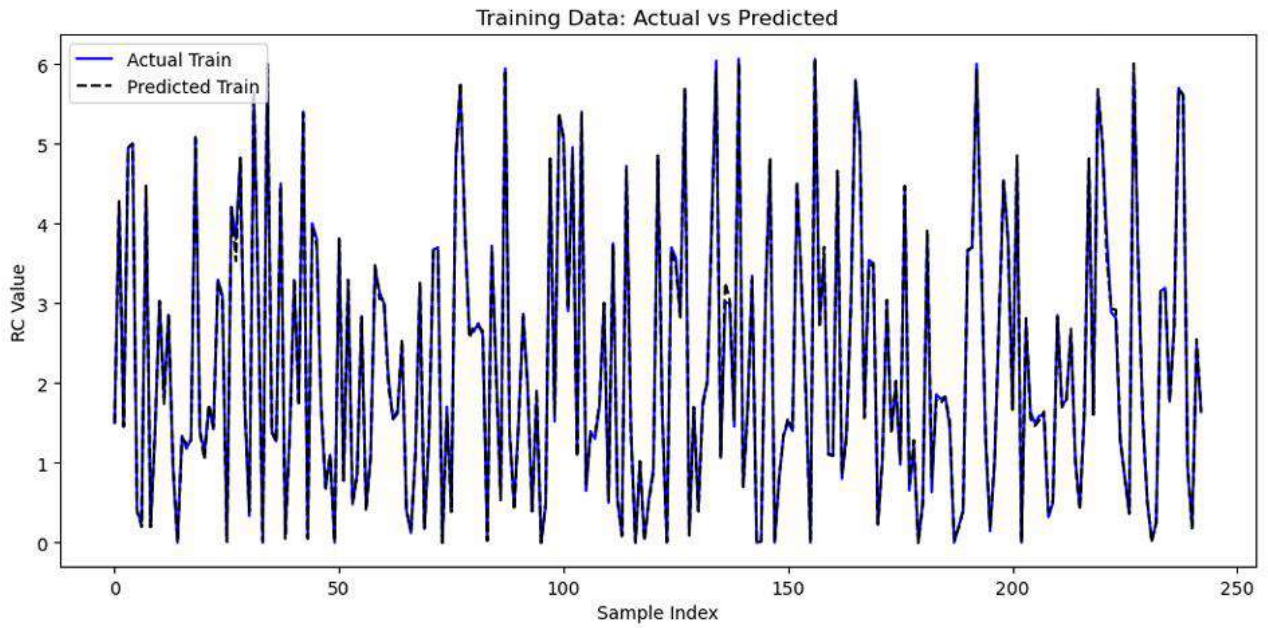


Figure (B.23): Comparison of Actual and Predicted Values for GRNN_D2_S2/Training.

Table (B. 23):Performance metric for GRNN_D2_S2/Training

GRNN-D2-S2	Training Results
MAE	0.036
MSE	0.002
RMSE	0.053
R^2	0.999

24. GRNN_D2_S3

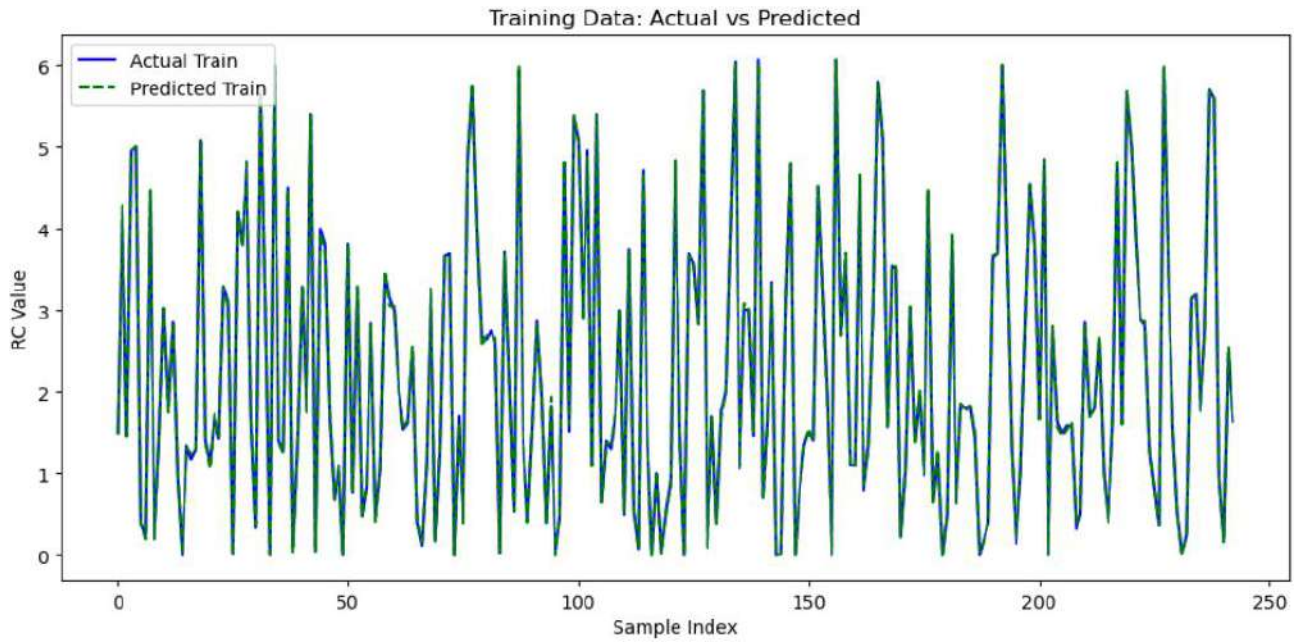


Figure (B.24): Comparison of Actual and Predicted Values for GRNN_D2_S3/Training.

Table (B. 24):Performance metric for GRNN_D2_S3/Training

GRNN-D2-S3	Training Results
MAE	0.029
MSE	0.001
RMSE	0.039
R^2	0.999

25. Transformer_D1_S1

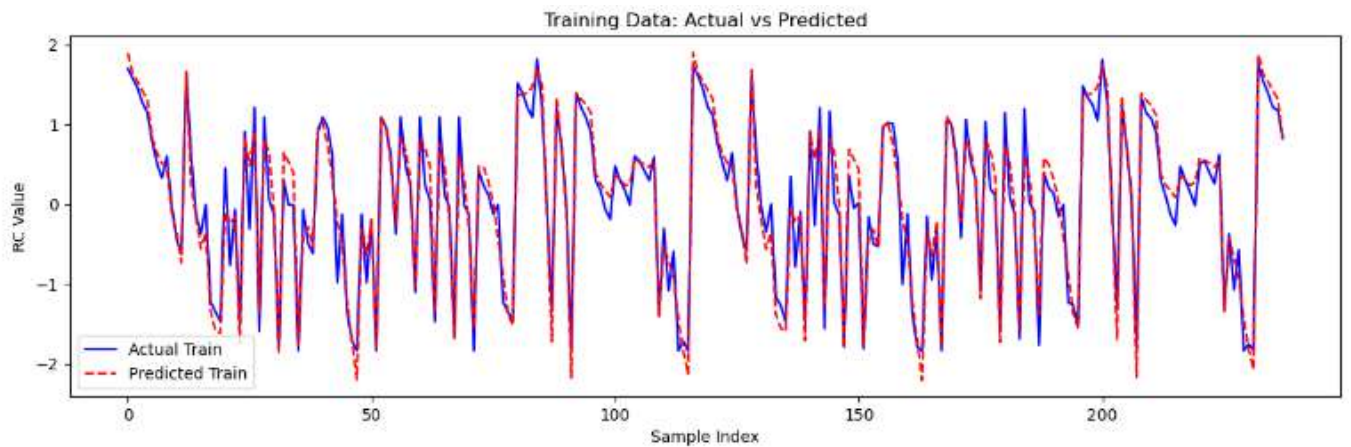


Figure (B.25): Comparison of Actual and Predicted Values for Transformer_D1_S1 /Training

Table (B. 25): Performance metric for Transformer_D1_S1/Training.

Transformer-D1-S1	Training Results
MAE	0.201
MSE	0.066
RMSE	0.258
R ²	0.934

26. Transformer_D1_S2

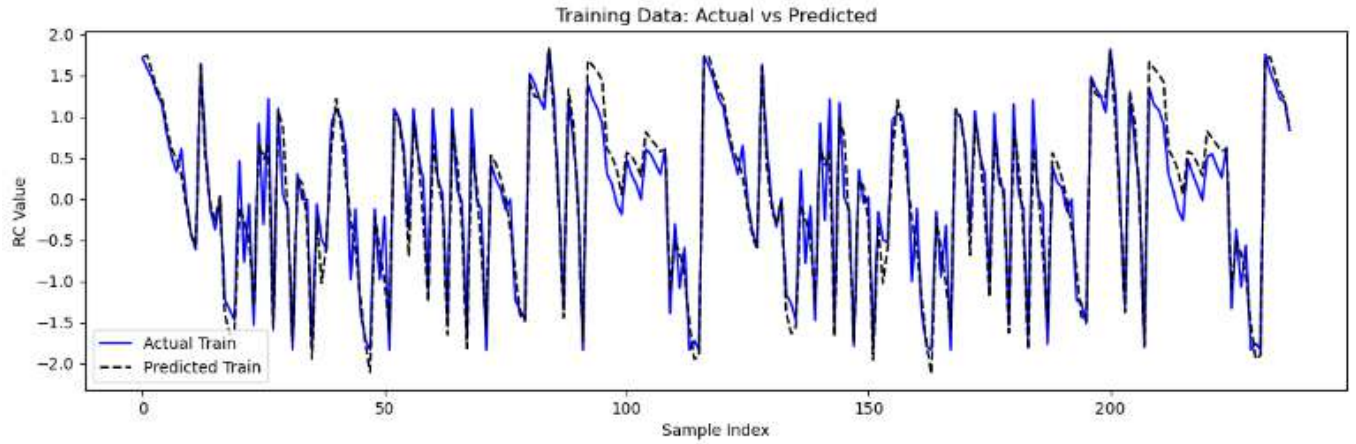


Figure (B.26): Comparison of Actual and Predicted Values for Transformer_D1_S2/Training.

Table (B. 26): Performance metric for Transformer_D1_S2/Training.

Transformer-D1-S2	Training Results
MAE	0.220
MSE	0.079
RMSE	0.281
R ²	0.922

27. Transformer_D1_S3

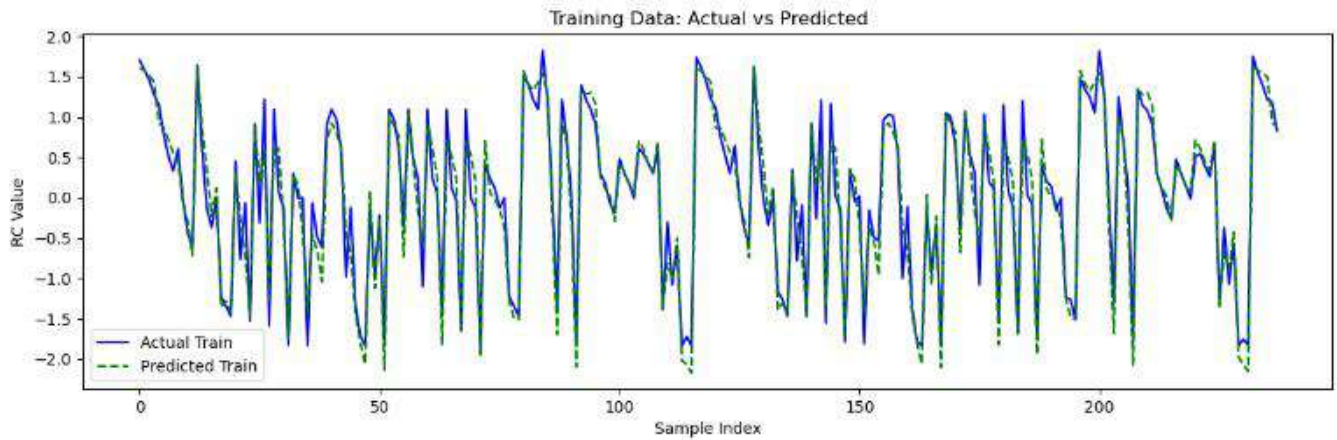


Figure (B.27): Comparison of Actual and Predicted Values for Transformer_D1_S3/Training.

Table (B. 27): Performance metric for Transformer_D1_S3/Training

Transformer-D1-S3	Training Results
MAE	0.197
MSE	0.065
RMSE	0.256
R ²	0.936

28. Transformer_D2_S1

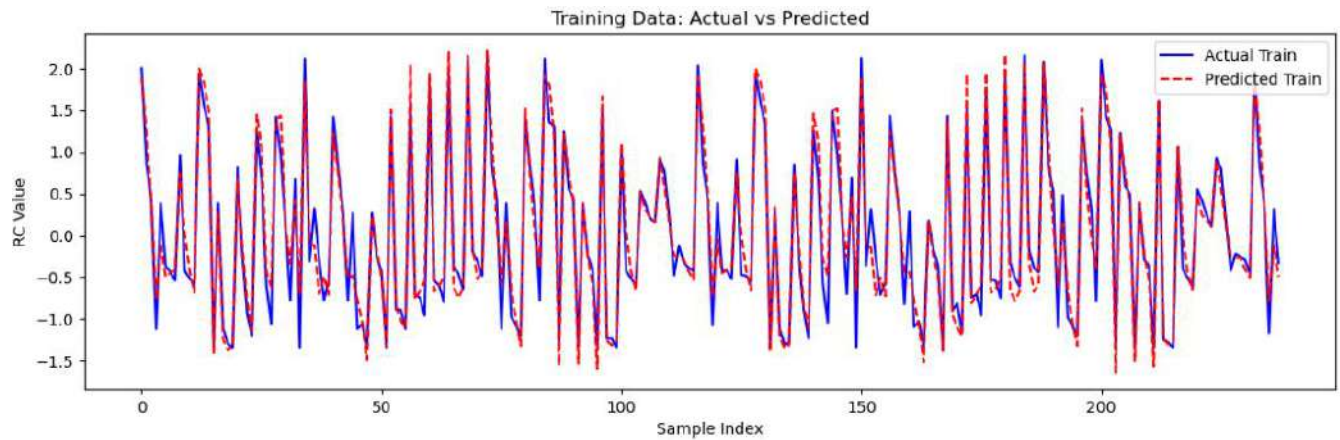


Figure (B. 28): Comparison of Actual and Predicted Values for Transformer_D2_S1/Training

Table (B. 28): Performance metric for Transformer_D2_S1/Training

Transformer -D2-S1	Training Results
MAE	0.183
MSE	0.058
RMSE	0.241
R ²	0.941

29. Transformer_D2_S2

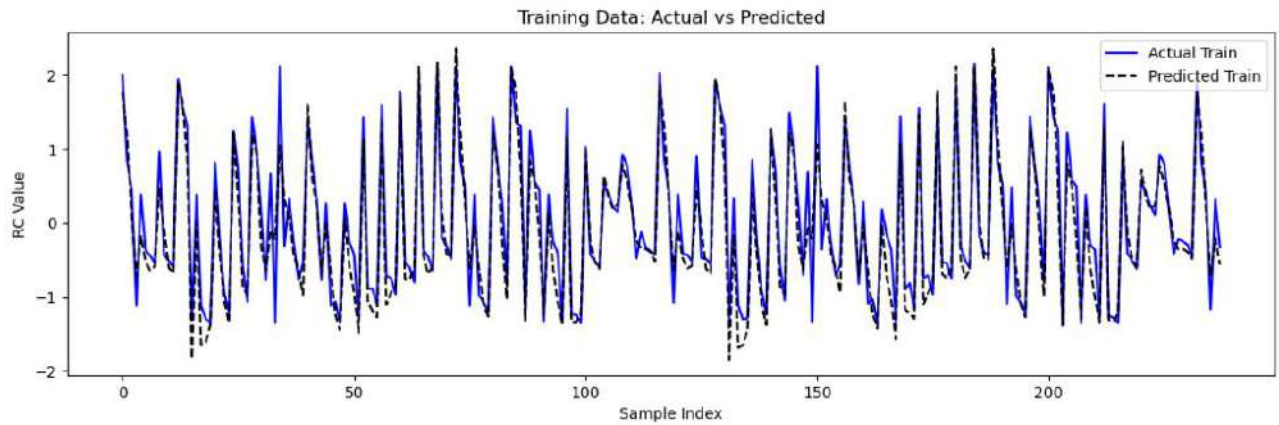


Figure (B. 29): Comparison of Actual and Predicted Values for Transformer_D2_S2/Training

Table (B. 29): Performance metric for Transformer_D2_S2/Training

Transformer -D2-S2	Training Results
MAE	0.245
MSE	0.104
RMSE	0.322
R^2	0.895

30. Transformer_D2_S3

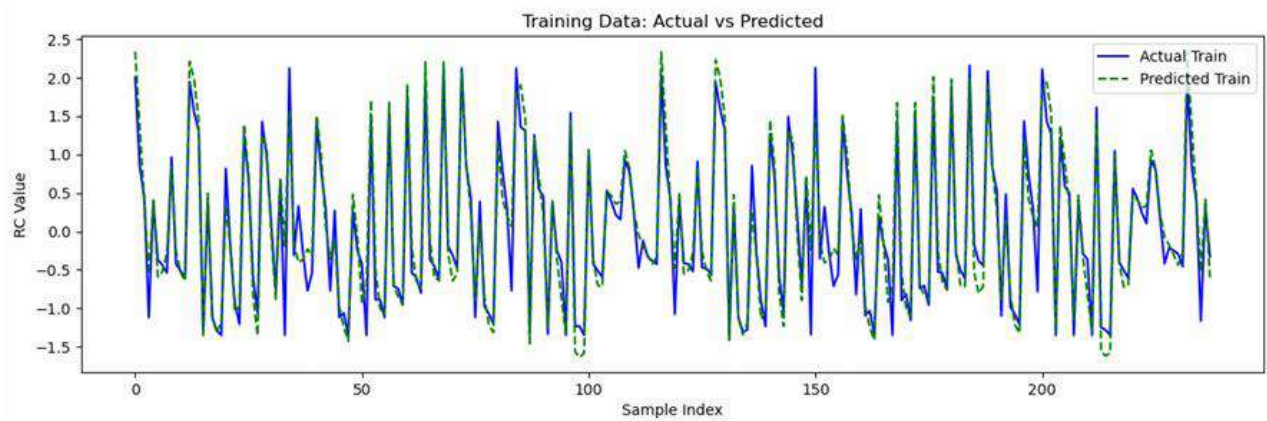


Figure (B. 30): Comparison of Actual and Predicted Values for Transformer_D2_S3/Training.

Table (B. 30 Performance metric for Transformer_D2_S3/Training

Transformer -D2-S3	Training Results
MAE	0.211
MSE	0.080
RMSE	0.283
R ²	0.919

الخلاصة

تُعد سلامة مياه الشرب أمراً حيوياً للصحة العامة، إذ يُشكل التطهير الفعّال محوراً أساسياً في تحقيقها. ويُعتمد على الكلورة على نطاق واسع لما توفره من أثر وقائي متبقٍ داخل شبكة التوزيع. غير أنّ الحفاظ على مستوى متبقٍ أمثل يُعدّ أمراً بالغ الأهمية؛ إذ تؤدي القيم المنخفضة إلى تقليل كفاءة التطهير، في حين أن المستويات المرتفعة تُسرّع من تآكل الأنابيب وتمثل مخاطر صحية. وقد تناولت هذه الدراسة نوعية المياه في محطة كربلاء المركزية (حي الحسين)، حيث يُستخدم الكلور للتطهير. ولغرض ذلك، جرى تدريب ثلاثين نموذجاً معتمداً على الذكاء الاصطناعي باستخدام خمسة خوارزميات، مع تقييم مدى مطابقتها لإرشادات منظمة الصحة العالمية (2017) والمواصفة العراقية لمياه الشرب رقم 417 (2009). كما أُجريت التحاليل الفيزيائية والكيميائية في المحطة وعلى طول خطي الأنابيب الرئيسيين. وأظهرت النتائج أن معظم المؤشرات كانت ضمن الحدود المقبولة، باستثناء العكارة والكلور المتبقي، وهو ما ارتبط بعدم كفاية معالجة العكارة في المحطة. ويمكن لمثل هذه الزيادات أن تؤثر سلباً في حالة الأنابيب داخل الشبكة وتُضعف فاعلية المُطهر.

فيما يخص نسبة الكلور المتبقي، فقد تجاوزت باستمرار الحدود المسموح بها في معظم النقاط تقريباً، الأمر الذي يسبب أضراراً تراكمية للمستهلكين فضلاً عن إلحاق الضرر بالأنابيب نتيجة تفاعله مع جدرانها. وتؤكد هذه النتائج على ضرورة قيام المحطة بإنشاء نظام رصد قادر على التحكم في مستويات الكلور واتخاذ قرارات استباقية بشأنها عند مواقع الرصد المحددة. أما من ناحية النمذجة، كانت الخوارزميات المستخدمة هي: ، (XGBoost)، (Extreme Gradient Boosting)، (General Regression Neural Network (GRNN)، (Multi Layer Perceptron (MLP)، و Transformer. وقد تم تدريب كل خوارزمية على ثلاثة سيناريوهات مختلفة باستخدام لغة Python اعتماداً على المدخلات.

نظراً لتقارب قيم معامل التحديد R وارتفاعها في جميع السيناريوهات، فقد تم اعتماد (MSE) Mean Square Error كمقياس أساسي للتقييم النهائي والمقارنة بين النماذج. وأظهرت نتائج الدراسة أن خوارزمية Transformer حققت قيم MSE ضمن المدى (0.053- 0.101) عبر الاتجاهات والسيناريوهات، في حين أظهر نموذج GRNN مدى (0.009- 0.219). وكلا النموذجين أبديا أداءً ممتازاً وتنافساً قوياً على المرتبة الأولى. أما المرتبة الثانية فقد حصل عليها نموذج XGBoost بمدى (0.102- 0.335)، تلاه RF بقيم تراوحت بين (0.197- 0.595). في حين جاء MLP في المرتبة

الأخيرة مسجلاً أعلى قيم MSE (0.625-0.117)، مما يعكس أضعف وأقل أداءً استقراراً عبر السيناريوهات المختلفة. خلصت هذه الدراسة إلى أن عملية تنقية المياه في المحطة بحاجة إلى تحسين، وأنه يمكن تطبيق نماذج الذكاء الاصطناعي الذكية كأداة داعمة في التنبؤ الاستباقي بمستويات الكلور المتبقي لتجنب أي أخطاء بشرية أثناء تطبيق جرعة الكلور.



جمهورية العراق
وزارة التعليم العالي و البحث العلمي
جامعة كربلاء
كلية الهندسة
قسم الهندسة المدنية

نمذجة وتوقع الكلور المتبقي لمياه الشرب باستخدام الذكاء الاصطناعي في مدينة كربلاء

رسالة مقدمة الى مجلس كلية الهندسة / جامعة كربلاء وهي جزء من متطلبات نيل درجة الماجستير في
علوم الهندسة المدنية

المؤلف:

نداء حبيب ابراهيم

بإشراف :

أ.د. سيف صلاح مهدي القزويني

أ.م.د. حيدر حسين علوان