National College of Ireland

Enhancing Water Use Data Analysis in Cloud Computing Environments through Parallel Processing Optimization

MSc Research Project
Cloud Computing

Priyanka Joseph
Student ID: x22114327@student.ncirl.ie

School of Computing
National College of Ireland

Supervisor: Ahmed Makki

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. ……Priyanka Joseph……

Student ID: ……x22114327@student.ncirl.ie……

Programme: ……Cloud computing…… **Year:** ……2023-2024……..

Module: ……MSc Research project……

Supervisor: ……Ahmed Makki……

Submission Due Date: ……14/12/2023……

Project Title: ……Ahmed Makki

Enhancing Water Use Data Analysis in Cloud Computing Environments through Parallel Processing Optimization……

Word Count: ……6600…… **Page Count**……23……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other

author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Priyanka Joseph……

Date: ……14/12/2023……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Water Use Data Analysis in Cloud Computing Environments through Parallel Processing Optimization

Priyanka Joseph
X22114327@studnet.ncirl.ie

Abstract

Water quality analysis is crucial for public health, ecological balance, and sustainable development. This research aims to perform water quality assessment using the National Water Quality Monitoring Programme (NWMP) dataset from India's Central Pollution Control Board (CPCB) ensuring the availability of safe drinking water and preserving water resources for ecological well-being. The significance is that predictive modeling and in the analysis is aimed at generating valuable information for data-driven decision-making regarding water resources by policy makers and scientists. The study utilizes PySpark, a Python-based Apache Spark framework, and machine learning algorithms on Amazon Web Services (AWS) SageMaker to process the large-scale dataset. The prime objectives are to effectively handle the expansive data, incorporate advanced analytics, and provide actionable insights for water resource management. The methodology contributes to the rearch by integrating PySpark for distributed data processing, applying linear and logistic regression models from Spark's machine learning library (MLlib) for predictive modeling, and leveraging AWS simple storage service (S3) for storage and AWS Glue for serverless integration. The study analyzes relationships in the parameters like dissolved oxygen, pH, conductivity to accurately estimate the Water Quality Index (WQI), which is an indicator of the consumable water quality. Linear regression model made predictions on the dataset and achieved a model accuracy of about 97%, while the logistic regression performed a litter better in classifying the water quality into multiple categories (Poor, Good, Unsuitable) with an accuracy of 99%. The findings will enable policymakers, water managers, and scientists to make informed decisions regarding sustainable water resource management. Overall, this research demonstrates a scalable, cloud-based approach combining PySpark, ML, and AWS for efficient large-scale water quality analysis.

Keywords: Water quality, CDWR, machine learning, PySpark, AWS

1 Introduction

Presently, there is a pervasive issue of diminished water quality that has far-reaching and consequential implications for several facets of human civilization, including but not limited to the economy, ecology, and the environment. This predicament has been linked to the emergence of diseases and the deterioration of human well-being (Kirschke et al.; 2020). In

rural regions and smaller localities, the potable water supply is typically sourced directly from wells or acquired from rivers, lakes, or reservoirs without sufficient treatment (Scheili et al.; 2015). Hence, the source of water significantly influences the overall quality of water. Therefore, it is imperative to comprehend the various aspects that influence the quality of water sources and to implement effective monitoring measures (Li and Wu; 2019). It is anticipated that the quantity of available data will increase in the future due to the process of digitization, leading to the development of digital services pertaining to the monitoring of water quality. These services aim to provide information regarding the suitability of water for drinking purposes. At present, the primary datasets utilized by water quality data services consist of network asset data, which is acquired from water bodies through the use of sensors, and water quality measurement data obtained from water-treatment plants as mandated by environmental authorities (Sirkia et al.; 2017).

The conventional approaches to evaluating water quality, which typically involve manual collection of samples and subsequent laboratory examination, encounter difficulties in managing the increasing amounts of data produced by contemporary monitoring systems. The emergence of big data technologies, such as frameworks like PySpark, offers a potential solution to address these limitations by facilitating the concurrent processing of extensive datasets. In light of these issues, the integration of sophisticated data processing technology and machine learning (ML) methods has emerged as a revolutionary strategy to address the intricacies embedded within extensive water quality datasets. This study undertakes a thorough investigation of water quality analysis, utilizing PySpark, a distributed data processing library, and machine learning algorithms implemented on the reliable infrastructure of Amazon Web Services (AWS). The objective is to utilize these technologies to examine the NWMP dataset, which contains extensive water quality data.

1.1Motivation

The standard methods employed for water quality analysis frequently encounter difficulties in managing the vast volumes of data produced by present monitoring devices. The NWMP dataset is widely recognized for its comprehensive representation of many water quality characteristics, rendering it a highly suitable choice for investigating big data analytics in order to reveal patterns and insights that may not be readily discernible using conventional approaches. Efficient processing is necessary to extract relevant information in an efficient way from the vast amount of data acquired from diverse water sources. PySpark, because to its distributed computing capabilities, is highly suitable for managing datasets of considerable size, hence facilitating expedited data processing and analysis. The utilization of machine learning approaches is an intriguing possibility for the development of prediction models pertaining to water quality metrics. The identification of trends, potential challenges, and the ability to make well-informed choices for the governance of water resources is of utmost importance. The utilization of PySpark on AWS enables efficient integration of distributed computing and machine learning, offering a comprehensive and effective approach for real-

time monitoring and decision-making in water quality management. This approach facilitates the development of predictive models for various water quality indicators.

1.2 Research Problem

The research problem pertains to resolving the scalability issue by conducting an experimental evaluation of the efficacy of PySpark, a distributed data processing framework, in handling extensive water quality datasets, such as the NWMP dataset. The process involves strategies to effectively expand the analytical capabilities in order to accommodate the increasing size of datasets, while also ensuring that the timeliness of the results prediction is not compromised. The study also aims to address the question of how machine learning algorithms, when combined with the PySpark processing pipeline, can effectively undergo training to construct prediction models for water quality metrics. The research inquiry pertains to the development of a comprehensive framework for the real-time processing of data on the Amazon Web Services (AWS) platform with the primary objective is to enable the water analysis pipeline to promptly identify changes in water quality parameters and provide timely information to facilitate effective decision-making.

1.3 Research Questions

This study aims to address the following primary research inquiries to comprehend the complex issues of water quality analysis.

Q: What approaches may be utilized for feature engineering to extract significant information from the various water quality parameters present in the NWMP dataset?

Q: How do machine learning and PySpark perform in terms of accuracy and efficiency compared to conventional approaches for analysing water quality?

1.4 Research Objective

The primary objective of this study is to build a comprehensive framework that effectively handles extensive water quality data, incorporates advanced analytics in conjunction with machine learning to estimate water quality parameters, and offer prompt, practical insights for water resources quality management. This research aims to realize this objective by utilizing PySpark and ML algorithms on the AWS platform to improve water quality analysis using the NWMP dataset.

1.5 Research Contributions

1) This study makes a valuable contribution by showcasing the efficacy of PySpark in addressing the scalability obstacles inherent in managing extensive water quality datasets. The inclusion of PySpark facilitates effective and parallelized data processing, enabling the examination of large datasets within a reasonable timeframe.

2) This research is focused on developing an approach that optimizes the efficient use of multiple parameters, thus providing a thorough understanding of water quality dynamics. This will be achieved by utilizing a wide range of water quality metrics available in the NWMP dataset.
3) This study offers a distinctive contribution by employing an extensive method in the selection and optimization of machine learning algorithms that are specifically tuned to accurately forecast various water quality characteristics. The research endeavors to enhance predictive abilities by training these models using historical NWMP data. This will facilitate the detection of long-term patterns and potential challenges relating to water quality.
4) This work makes a distinct contribution by exploring and improving the inclusion of several AWS services into the pipeline for water quality analysis. This entails utilizing cloud computing resources to optimize the performance, cost-efficiency, and adaptability of the framework.

1.6 Limitations of the study

This study is limited to the scope of examining the WQI of the Indian water bodies like lakes, canals and rivers gathered from NWMP dataset which is restricted due to the geographical or regulatory constraints. The study could also make use of advanced ML and deep learning models for time series forecasting for more comprehensive analysis of the water quality parameters.

1.7 Thesis structure

This research paper is divided into five main sections:
Section 1: The section outlines our research background and why we conducted further study. Additionally, it gives the theme of investigations as well as responding to the research question.
Section 2: This part presents a critical review of the previous studies that have been developed for years.
Section 3: Provides an elaborate account of research methodology in order to enhance understanding of the topic under review.
Section 4: This section outlines the elaborate procedures and the techniques utilised to implement the proposed strategies. It provides information about the instruments applied including data sources, and also explains how the performances of the results would be assessed.
Section 5: This section presents an in-depth analysis of the study's main findings and their study's results.
Section 6: This section provides a summary of the research project that includes techniques employed, ways through which these techniques were performed and results generated by the process. It also makes recommendations for further studies.

2. Literature Review

The study of water quality has received considerable focus in the field of environmental science because of its crucial implications for public health, ecological equilibrium, and sustainable progress. Advanced technology, machine learning algorithms, and big data analytics have enhanced and, in certain instances, substituted traditional approaches to water quality evaluation. This literature review examines prominent research, methodologies, and technologies associated with water quality analysis, with a specific emphasis on the utilization of PySpark and machine learning techniques on AWS. It investigates the applications and contributions of these approaches in the field of environmental science.

2.1 Big Data in Water Quality Analysis

Hemdan et al. (2021) discuss the use of IoT and big data analytics in water quality monitoring. The paper highlights the importance of preserving and taking care of water resources for human survival and access to clean and safe water resources. The study presents a review of smart water quality analysis, which can help in developing an agile environment that can handle the massive flow of water big data generated by smart sensors. The paper provides an overview of the system components, applications, essential parameters, and big data analytics workflow in water quality monitoring. (Nair and Vijaya; 2021) presented the importance of monitoring and predicting the quality of river water due to pollution caused by human and industrial waste. Traditional methods for evaluating water quality have limitations, and big data techniques have been used to develop predictive models. They review different prediction models and their experimental results, as well as challenges and proposed solutions. The paper also discusses water treatment, water quality standards, and computational methods of various research works in water quality evaluation and prediction were also reviewed.

Nie et al. (2020) discussed the use of Big Data analytics and IoT in operation safety management in underwater management, specifically in water conservation and management. The Supervisory controller and data acquisition (SCADA) Approach for sustainable water management in the smart city based on IoT and Big Data Analytics was proposed using a mathematical model. Overall, it was concluded that the use of IoT and Big Data can enhance water management in several ways, including water leak identification, water quality monitoring, and infrastructure maintenance. (Berlian et al.; 2016) proposed the development of a smart environment system for monitoring water quality and coral reefs in rivers, using the Internet of Underwater Things and big data. The system framework consists of remotely operated vehicles (ROVs) equipped with water quality sensors, portable water quality monitoring systems, coral reef monitoring systems, wireless mesh networks, and big data analytics. The collected data is analyzed and visualized on an open platform based on a Hadoop Multi-Node Cluster and can be accessed globally.

Kimothi et al. (2022) proposed a big data analytics framework to analyze water quality parameters in Uttarakhand and concluded that the water is safe to drink except for in the Haridwar district where there is an increase in contaminants. The study used statistical and fractal methods to analyze the anomalies between the water quality parameters in 13 districts

of Uttarakhand. The mean, mode, standard deviation, median, kurtosis, and skewness of time series datasets were examined. The variation in WQP was analyzed using a random forest (RF) model. The dataset was segmented location-wise and the mean, mode, standard deviation, median, kurtosis, and skewness of time series datasets were examined. The water samples were found to be safe to drink and in healthy condition in almost all the districts of the state Uttarakhand, except for the Haridwar district, where some increase in contaminants was observed. (Han et al.; 2022) presented the development of a web-based platform for analyzing water quality using big data analysis that integrates both traditional methods and big data methods to evaluate water quality. The platform provides an open interface to access water quality data from sensors and can automatically import data and can provide users with comprehensive analysis results in both text and graphic format, which can be easily understood by the user using web-based framework.

2.2 Big Data Analytics using Hadoop MapReduce

A distributed framework for large-scale protein-protein interaction data analysis and prediction using MapReduce is proposed by (Hu et al.; 2021). The framework aims to overcome the limitations of existing algorithms in predicting protein-protein interactions (PPIs) by using genomic information, evolutionary profiles, and protein sequences. The proposed framework, CoFex+, is a modified version of CoFex and uses a MapReduce framework to significantly improve efficiency when applied to large-scale PPI prediction. The authors conducted extensive experiments to demonstrate the promising accuracy of CoFex+ and addressed the efficiency bottlenecks of CoFex in a distributed manner. (Chiang et al.; 2021) documented the use of Petri nets (PN) to model and analyze Hadoop MapReduce systems for big data. The MapReduce framework is a core component of Hadoop, used for parallel computing in big data analysis. The study presents a Petri net model to describe the internal procedure of the MapReduce framework, common errors, and an error prevention mechanism using the PN models to increase efficiency in system development. The study demonstrates the feasibility of the PN model and its potential to assist in developing parallel MapReduce systems.

Ramani et al. (2020) presented a modified artificial neural network (ANN) classifier technique with a MapReduce framework for the prediction of diabetic chronic disease. The proposed system aims to provide accurate, fast, and optimal results on chronic disease datasets. The experimental results over chronic diabetic dataset prove that the proposed ANN with MapReduce structure is capable of predicting the precision, sensitivity and specificity level modified on comparing with other existing deep neural network approaches. The proposed methodology involves normalization, MapReduce algorithm, and modified ANN classifier on MapReduce. Wen et al. (2020) stated the use of a MapReduce-based back-propagation (BP) neural network for the classification of aquaculture water quality. The BP neural network is a computer-generated network that simulates human brain and responses. The MapReduce processing model is used to perform parallel design of the BP neural network algorithm to meet the needs of massive data processing in aquaculture platforms. The optimized BP neural network algorithm is used to provide efficient, fault-tolerant aquaculture data management,

mining, and visualization. (Ma et al.; 2020) proposed novel approaches to deal with the scalability and performance problem of similarity join query on massive high-dimensional data sets. Three methods based on projection scheme are proposed and evaluated through experiments using MapReduce to improve efficiency. The approaches show good performance and scalability. The paper also surveys related works on similarity join and introduces relevant theorems.

2.3 Big Data Analytics using Apache Spark and ML

Meti and GK. (2020) analyzed the importance of rainfall analysis in agriculture and the various methods of rainfall prediction that have been developed over the years. The authors proposed a new predictive algorithm for future rainfall prediction that combines various AI and ML techniques that can be used to mine patterns in rainfall data. The research also provided a literature survey of recent literature in this domain and a comparative study of rainfall analysis using big data analytics using various methods including PySpark, and Hive. (Ranganathan. 2020) proposed a real-time anomaly detection system using PySpark to identify abnormalities in cloud data centers with multiple-source VMware. The procedure utilizes PySpark to compute data batches, with minimized delay, and a flat-incremental clustering to frame the normal attributes in the PySpark Structure. The latencies in computing the tuple while clustering and predicting were compared for PySpark, Storm, and other dispersed structures used in processing the batches of data. The processing time of a tuple in PySpark was much less compared to other methods.

Alherbi et al. (2022) presented a study on predicting oil production values using linear regression and big data tools. The authors proposed a method that takes into account various independent variables, such as pressure, downhole temperature, and pressure tubing to accurately predict the actual production value. The study highlights the significance of accurate oil production prediction and the potential of machine learning algorithms in solving issues in almost all areas of the oil industry. (Krishnan et al.; 2021) implemented sentiment analysis of COVID-19 related Twitter data using ML techniques. Sentiment analysis is used to determine the polarity of people's sentiments towards COVID-19 situations, which can help authorized bodies to take suitable actions. The study uses classification algorithms such as logistic regression, decision tree, random forest, and Naive Bayes for sentiment analysis. The tweets are extracted from Twitter as Dstreams using PySpark streaming, and the data is fitted to the machine learning pipeline of the model.

Choumal and Yadav. (2021) proposed the use of ML approaches for fault detection in photovoltaic (PV) arrays using PySpark. The study focused on feature extraction from I-V curves under various fault occurrences and standard conditions. The ML library in PySpark is used to examine these attributes and detect faults. The paper also compared several classification methods using a confusion matrix addressing soft accuracy, precision, and recall. Ultimately, the study deployed a classification model using PySpark and MLlib to categorize faulty and normal conditions and enumerates various performance metrics. (Semic and Karamehic; 2022) proposed using Support Vector Machine (SVM) and Random Forest (RF)

Regression, to predict the likelihood of stroke and heart attack based on physiological parameters. PySpark was used to process and analyze a large dataset, and the results show high accuracy scores for both models. The research also analysed the causes and prevalence of stroke and heart attacks, as well as related studies on using ML based disease prediction. A study that uses an automated ML pipeline (AutoML) to find the best model for predicting total nitrogen concentrations in the Chesapeake Bay watershed was presented by (Kim et al.; 2020). The highest performing model was a stacked ensemble model, outperforming deep learning models. The most important predictors for predicting nitrogen concentration were geographic indicators. The research found that location and land use were important predictors, while climate variables were poor predictors. The highest performing model was a stacked ensemble model, H2O AutoML Regression GLM, with an R2 of 0.91 and RMSE of 0.48 on the test data. The study also examined the combined effects of land use and climate change on nutrient concentrations for the Chesapeake Bay.

2.4 Big Data Analytics on AWS Cloud

Vilas et al. (2020) developed 1622WQ, a web-based tool to raise farmer knowledge of agriculture's impact on AWS cloud water quality. The program addresses adoption hurdles such restricted internet connection, low data quality, and operational concerns utilizing user-centered design. The research focused on raising farmer understanding of agriculture's impact on water quality and encouraging voluntary improvements. Research suggests that farmers may not comprehend the link between crop management and water quality and may not realize their role to water quality decline. (Albaldawi et al.; 2022) suggested using big data analytics in healthcare, specifically hybrid minhash models with four classification methods for sentiment analysis on large datasets like COVID-19 Vaccine Stance tweets and Covid-19 Tweets IEEE data port datasets using Apache Spark data. The suggested model classified tweets using several classifiers with excellent accuracy, proving that Spark models can process large-scale data. The study also indicated that customers in regulated industries like healthcare choose Amazon EMR for data protection. Apache Spark and ML for healthcare big data analysis were also discussed.

Mareeswari et al. (2021) proposed the use of Apache Spark and Scala for real-time sentiment analysis of tweets using an ML based text classification algorithm and recommended preprocessing for better results. The system was implemented on AWS and featured a dashboard for users to see a summary of sentiment analysis of all tweets, a detailed analysis of tweets, and an ad-hoc run for analyzing the sentiment of text before posting to digital media. (Kılınç. 2019) presented a framework for real-time sentiment prediction on streaming data using Apache Spark's ML and streaming service on AWS. The proposed system consists of four integrated software components, including ML and streaming service for sentiment prediction, a Twitter streaming service to retrieve tweets, a Twitter fake account detection service to assess the owner of the retrieved tweet, and a real-time reporting and dashboard component to visualize the results of sentiment analysis. The sentiment classification

performances of the system for offline and real-time modes are 86.77% and 80.93%, respectively.

Seal and Mukherjee. (2019) suggested the use of Spark streaming on Amazon's Elastic Mapreduce (EMR) to provide real-time cloud-based analytics of live video feeds from the cameras of self-driven autonomous vehicles. The methodology used deep-learning methodologies for real-time object detection on the streamed images to classify and predict traffic incidences leading to subsequent congestion control. The results showed a 60% improvement in performance on the AWS-EMR based Spark framework when compared to cloud processing on a single instance of EC2 server on the AWS. (Hafez et al.; 2023) proposed the use of a modified decision tree (DT) algorithm for analyzing big data. The study also presented a framework for improving the quality of prediction, which includes data extraction, processing, and evaluation measures. The proposed model was evaluated using different evaluation metrics, including recall, precision, accuracy, and F1-score. The study concludes that the modified DT algorithm shows better accuracy in handling different-sized datasets compared to the standard DT algorithm. The study involved a comparative analysis conducted on Apache PySpark, utilizing a multi-node setup on AWS.

2.5 Summary

From the literature discussed, it can be summarised that the focus on integrated frameworks for large-scale water quality analysis was limited and there is a necessity for an end-to-end model that bring together data processing, ML models and cloud infrastructure for scalable and efficient water quality assessment. There was also a shortage of evaluations on real-world datasets like NWMP used in this research. This study proposes to develop an end-to-end big data framework integrating PySpark, ML models, and AWS cloud platform for scalable water quality examination using real-world NWMP data. The future research directions can lead to exploring advanced ML models like XGBoost, developing stream processing capabilities for real-time analytics and testing with different water quality datasets.

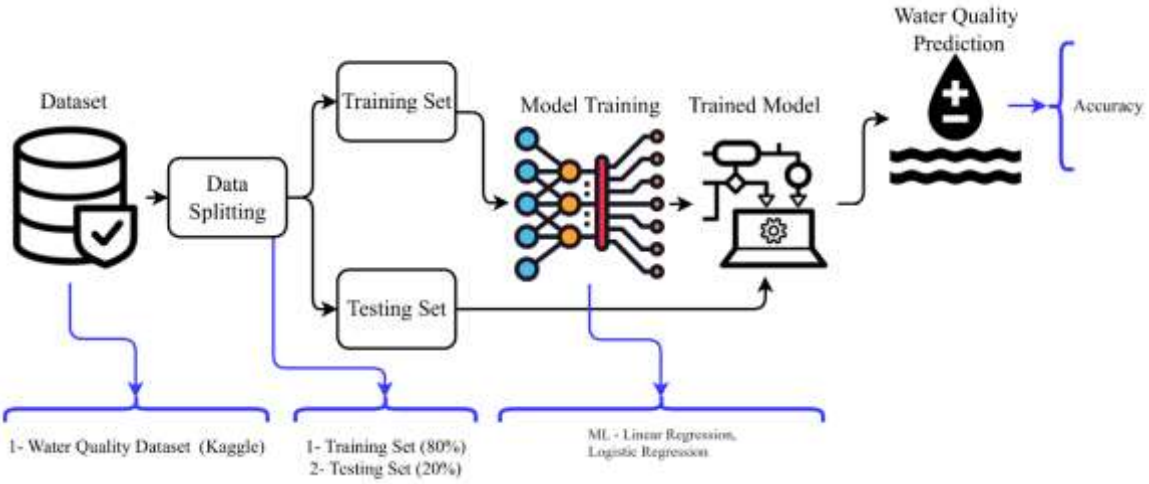
3. Research Methodology

This section provides a comprehensive explanation of the research technique. Figure 1 illustrates the research technique, which includes data collecting, data pre-processing, model training, and model evaluation.

3.1 Data collection: The data was collected from the official website of the Central Pollution Control Board (CPCB) of the Indian government, specifically from the National Water Quality Monitoring Network Programme (NWMP)[1]. The purpose of gathering this data was to evaluate the current condition of water quality in various water resources and to aid in the prevention and management of water pollution. The monitoring network consists of 4484

[1] https://cpcb.nic.in/nwmp-data/

stations that track surface and groundwater in 28 States and 8 Union Territories. Monitoring is conducted at regular intervals, including monthly, quarterly, semi-annually, and annually. The water samples undergo analysis for 9 fundamental factors and are then compared to the recommended water quality requirements for optimal use as suggested by CPCB. The dataset was obtained from the Kaggle website[2] and contains important parameters such as dissolved oxygen (DO), pH, conductivity, biological oxygen demand (BOD), nitrate, fecal coliform, and total coliform. This information is utilized to guarantee the excellence of the provided potable



water. The water quality index (WQI) and the quality state of water can be forecasted based on these criteria. Data preparation, which involves the computation of Water Quality Index (WQI) and the categorization of water samples based on their WQI values, is a crucial stage in the study.

Figure 1. Research Methodology

3.2 Data Preprocessing: Data preprocessing is an essential component of this methodology, and it entails the following procedures for preprocessing the NWDP dataset: The dataset will be divided into two segments: a training set (80%) and a testing set (20%). It is customary to train the model on one portion of the data and then assess its performance by evaluating it on a different portion. Data cleaning encompasses the process of discovering and addressing missing values, outliers, and errors within a dataset. Depending on the characteristics of the data, this process may require the removal or imputation of missing values. Only the pertinent characteristics that are likely to have an impact on both the quality and usage of water were selected. To enhance model performance and minimize complexity, extraneous or repetitive features were eliminated. The data needs to be converted into a format that is appropriate for ML models. These involve the normalization or standardization of numerical values, the encoding of categorical variables, and the creation of new features through feature engineering. Categorical variables were transformed into a numerical format **u**sing approaches such as one-hot encoding or label encoding, depending on the specific requirements of the algorithm being used.

[2] https://www.kaggle.com/datasets/akkshaysr/nwmp-water-quality-data-for-indian-lakes/data

3.3 Model Design and Training: The ML model is trained using a preprocessed dataset that includes specific attributes. ML classifiers utilized for training are linear regression (LR) and logistic regression (LrR). The classifier models are trained using all available features, and the Python module, Scikit-learn is used to develop and test the models in this study. The model is trained using the gathered dataset on water quality to predict the WQI based on the various chemical compositions found in the water samples. LR and LrR are distinct statistical techniques employed for prediction, each suited for different categories of situations. Linear regression is a statistical model used to forecast a continuous outcome variable, which is typically referred to as a regression problem. It is suitable when you desire to forecast a numerical value, such as the WQI, which usually spans throughout a continuous range of values. Logistic regression is primarily employed for solving classification problems, though the name would suggest a regression model It is used when the outcome variable is categorical, such as classifying WQI into categories like 'Good', 'Moderate', or 'Poor'. The model provides an estimation of the likelihood that a specific input value is a member of a particular class. The output is subjected to a logistic function, commonly referred to as the sigmoid function, in order to guarantee that the resulting values fall within the range of 0 and 1.

3.4 Model Evaluation: Model evaluation is an essential stage in ML pipeline process, as it allows for the assessment of model performance and its capacity to generalize to unfamiliar data. The evaluation method for predicting WQI using linear and logistic regression on the NWDP dataset would vary depending on the type of regression employed. Both methods of regression utilized a distinct testing set to assess the model's performance. The use of this testing set throughout the training phase was avoided in order to guarantee that the evaluation accurately represents the model's capacity to apply its knowledge to unfamiliar data. The performance metrics will be computed to assess their predictive capacity for water quality forecasting, using the accuracy percentage as the evaluation criterion.

4. Design Specifications

The design specifications for predicting WQI using linear and logistic regression on the NWDP dataset hosted on AWS Sagemaker is discussed as follows:

4.1 Linear Regression: Linear regression is a statistical technique used to model the association between a dependent variable X and one or more independent variables (Y). The equation for a linear regression model with one independent variable can be expressed in the general form as follows:

$$\boldsymbol{Y = \beta_0 + \beta_1 X + \varepsilon}$$

β_0 is the intercept and β_1 is the coefficient for the independent variable. ε is the error term, that captures the variability in the dependent variable not explained by the independent variable. The objective of linear regression is to calculate the coefficients that minimize the sum of the squared deviations between the actual and projected values of the dependent variable.

4.2 Logistic Regression: Logistic regression is a statistical model employed for binary classification tasks, aiming to forecast the probability of a given input belonging to a specific class. Logistic regression employs the logistic function to mathematically model the association between the input features and the probability of the output. The logistic regression model can be expressed mathematically as:

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p)}}$$

$P(Y = 1 \mid X)$ denotes the likelihood that the output (Y) is equal to 1, given the input features (X). $\beta_0, \beta_1, \beta_2, .. \beta_p$ are the coefficients or weights associated with the input features. $X_1, X_2, .. X_p$ are the input features. The logistic function, also known as the sigmoid function, is used to map the linear combination of the input features and their associated weights to a value between 0 and 1, representing the probability of the output belonging to a particular class. In logistic regression, the primary goal is to determine the optimal values for the coefficients β that maximize the likelihood of the observed data.

4.3 Data Analytics on PySpark and AWS Sagemaker: PySpark is the amalgamation of Python with Apache Spark, a robust distributed computing technology that is open-source. The Python API enables the user to harness the functionalities of Spark. PySpark is a useful tool for effectively processing and analyzing large-scale water quality datasets in the field of water quality study. It enables activities such as data preprocessing, feature engineering, and machine learning model creation. This was done by utilizing AWS Glue, which offers help for PySpark in processing large volumes of data. PySpark was utilized for data preparation, while AWS SageMaker was employed for both model training and deployment. SageMaker offers a smooth connection with PySpark for efficient handling of extensive data processing and ML assignments. The design process to work with PySpark on the NWMP water quality dataset, in Figure.2, involves the following key steps:

1. Utilize PySpark to ingest the extensive dataset into a distributed data structure, such as a Resilient Distributed Dataset (RDD) or a DataFrame. This phase is crucial for the implementation of parallel processing and distributed computing.
2. To prepare the big dataset for analysis, perform data preparation operations like feature engineering, data transformation, and data cleaning. PySpark offers a range of methods and capabilities to effectively preprocess distributed data.
3. Employ PySpark to conduct data analysis on the NWMP dataset, encompassing statistical analysis, aggregations, and training the ML model for WQI prediction. PySpark's ML package, MLlib, offers scalable techniques for training and evaluating models.

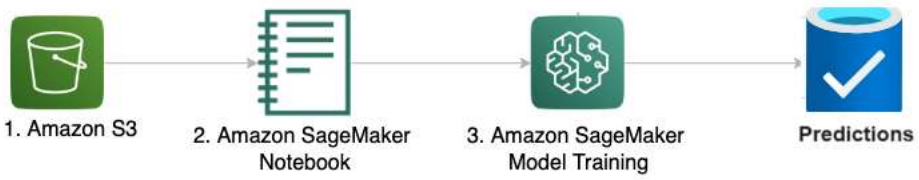


Figure 2. AWS PySpark Implementation Process Tree

5. Implementation

The implementation process for this research consists of uploading the water quality dataset to the AWS S3 storage, followed by accessing within the PySpark environment created using a notebook instance in Amazon SageMaker. This section explains the process in detail.

The main steps involved in the implementation are:

1. Preprocess and prepare water quality data
2. Train logistic and linear regression models with PySpark
3. Make predictions on new test data
4. Evaluate model accuracy

5.1 S3 Storage setup and access: It is imperative to verify that Sagemaker possesses the necessary authorization to access S3 storage and get the data first. Setting up an IAM role for the Sagemaker notebooks is necessary in order to grant them permission to access and read data from S3 buckets. Then, we can setup the s3 bucket storage and upload the file as in Figure 3.

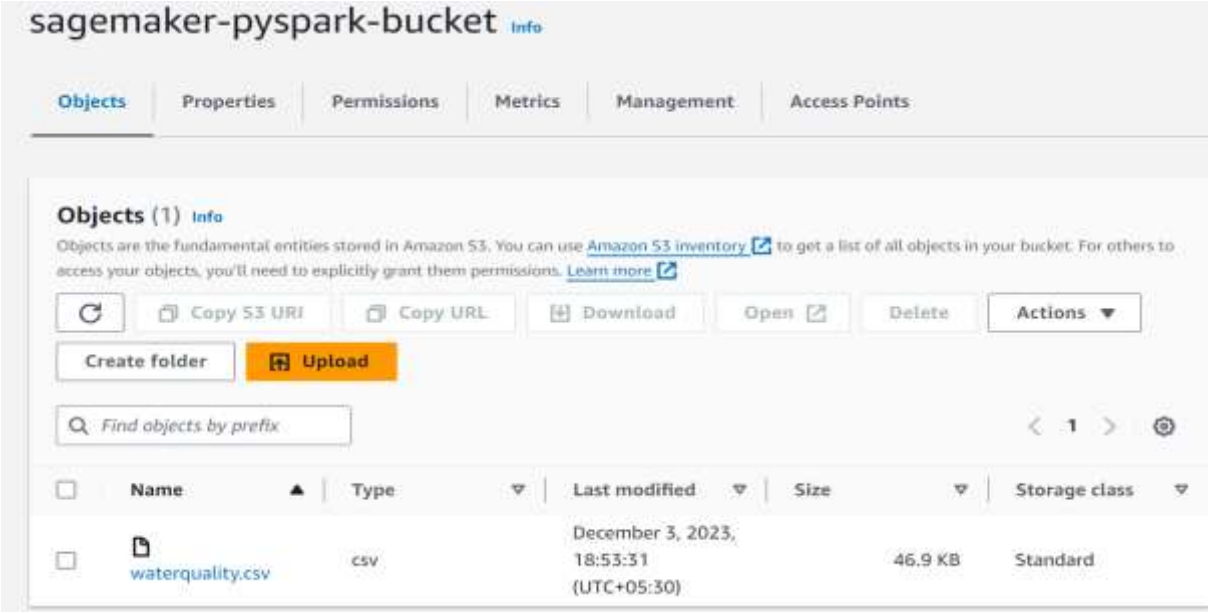


Figure 3. S3 bucket storage with waterquality.csv file uploaded

5.2 Sagemaker IAM policy and role creation to access AWS Glue: AWS Glue is a serverless data integration solution offered by AWS. This tool streamlines the procedure of identifying, organizing, combining, and updating data for the purposes of analytics, ML, and application development. AWS Glue is a service that does ETL (extract, transform, load) operations on data, automatically analyzing and categorizing it.

The following are the steps involved in configuring IAM permissions for AWS Glue.

1. Create an IAM policy for AWS Glue service that allows access to S3 storage, EC2 instances, writing to CloudWatch metrics etc.,
2. Establish an IAM role for AWS Glue to confer rights to your IAM role that AWS Glue can adopt when making requests to other services on your behalf. This encompasses the ability to utilize Amazon S3 for all sources, targets, scripts, and temporary directories that are employed in conjunction with AWS Glue.
3. Attach the following policies to users or groups that access AWS Glue, GlueConsoleAccessPolicy, GlueAWSGlueConsoleSageMakerNotebookFullAccess AWSCloudFormationReadOnlyAccess.
4. Generate an IAM policy specifically designed for notebook servers. This policy authorizes certain Amazon S3 activities to oversee the resources in your account that are required by AWS Glue when it assumes the role specified in this policy.
5. Create an IAM role for notebook servers that AWS Glue can adopt when making requests to other services on your behalf. This encompasses the ability to utilize Amazon S3 for all sources, targets, scripts, and temporary directories that are employed in conjunction with AWS Glue.
6. Finally, Create an IAM role for SageMaker notebooks.

5.3 PySpark Enviroment Creation:

The SparkSession serves as the primary interface for programming Spark using the Dataset and DataFrame API. This feature enables the user to generate and set up a SparkSession with a range of customizable settings. The interface serves as a centralized access point for interacting with the underlying Spark functionality and enables programming Spark using DataFrames. SparkConf is utilized to setup the Spark application. This function is utilized to assign different Spark parameters as key-value pairs. The purpose of these configuration parameters is to instantiate a SparkConf object, which is subsequently utilized to instantiate a SparkContext.

```
from pyspark import SparkConf
from pyspark.sql import SparkSession
import sagemaker_pyspark
import botocore.session

session = botocore.session.get_session()
credentials = session.get_credentials()

conf = (SparkConf()
        .set("spark.driver.extraClassPath", ":".join(sagemaker_pyspark.classpath_jars())))

spark = (
    SparkSession
    .builder
    .config(conf=conf) \
    .config('fs.s3a.access.key', credentials.access_key)
    .config('fs.s3a.secret.key', credentials.secret_key)
    .appName("schema_test")
    .getOrCreate()
)
```

Figure 4. Importing Libraries - Create SparkSession with the user access key id & secret key id

The botocore.session module enables the creation of a Session object, which serves as a container for configuration settings and facilitates the creation of service clients and resources. The botocore.session.Session class can be utilized to instantiate a service client at a low level, manage event handlers, and execute many associated operations. Refer Figure 4 for the code level implementation of this module.

5.4 WQI Calculation:

The Water Quality Index (WQI) is a metric utilized to quantify the comprehensive quality of water, taking into account multiple characteristics. The calculation of the WQI requires four primary steps: parameter selection, sub-index determination, weight assignment, and final index aggregation. The exact method for calculating the WQI may differ depending on the unique situation and the organization employing it. The calculation of the WQI typically entails assigning weights to various water quality parameters, including Dissolved Oxygen (DO), Biochemical Oxygen Demand (BOD), Chemical Oxygen Demand (COD), Suspended Solids (SS), pH, and Ammonia (AN). These weighted parameters are then combined into a single index that represents the overall water quality.

It is crucial to acknowledge that the relative weights allocated to the factors in the WQI formula might be subjective and prone to variation based on expert judgments or the unique circumstances in which the index is applied. Objective techniques, including the Bayesian model-based approach, have been implemented in certain circumstances to achieve more consistent and unbiased relative weights for the factors utilized in the production of the WQI. The calculation of WQI entails the assessment of various water quality parameters, the allocation of weights to these parameters, and the consolidation of these weighted parameters into a unified index. The precise formula and procedures for calculating the WQI are outlined below.

$WQI = \sum (qn \times Wn)$

where qn =Quality rating for the nth Water quality parameter, Wn= unit weight for the nth parameters.

5.5 Psuedocode for Linear and Logistic Regression:

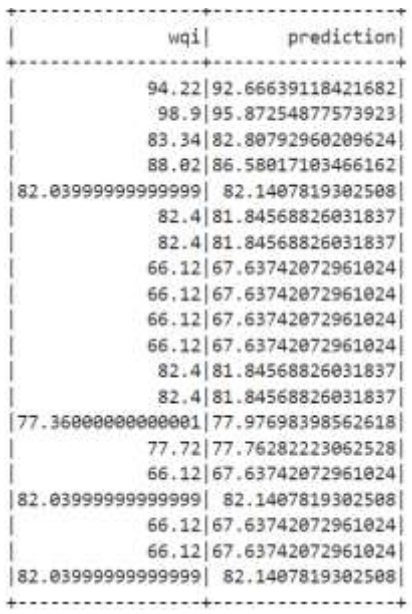
```
# Import libraries
import pyspark
from pyspark.ml import Pipeline
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.regression import LinearRegression
from pyspark.ml.feature import VectorAssembler
# Load data
df = spark.read.csv("s3://bucket/waterquality.csv")
# Preprocess data
```

```
df = handle_missing_values(df)
df = encode_categorical_features(df)
df = normalize_features(df)

# Train test split
train_df, test_df = df.randomSplit([0.8, 0.2])

# Feature vector
assembler = VectorAssembler(inputCols=["pH", "turbidity", "DO"], outputCol="features")

# Logistic regression
logreg = LogisticRegression()
logreg_pipeline = Pipeline(stages=[assembler, logreg])
logreg_model = logreg_pipeline.fit(train_df)

# Linear regression
linreg = LinearRegression()
linreg_pipeline = Pipeline(stages=[assembler, linreg])
linreg_model = linreg_pipeline.fit(train_df)

# Make predictions
logreg_predictions = logreg_model.predict(test_df)
linreg_predictions = linreg_model.predict(test_df)

# Evaluate models
print(evaluate(logreg_predictions, test_df))
print(evaluate(linreg_predictions, test_df))
```

6. Experimental Results

6.1 Case-1: WQI prediction using Linear Regression

Linear regression is used as a predictive modeling approach to estimate WQI based on various water quality predictor variables in this study. The model is evaluated by comparing the actual WQI values with the predicted WQI values as shown in Figure.5. From the figure, the performance of the linear regression-based model is significantly closer to the actual WQI predictions asserting how well the model performed. The model classification accuracy was 97.38%.



wqi	prediction
94.22	92.66639118421682
98.9	95.87254877573923
83.34	82.80792960209624
88.02	86.58017103466162
82.03999999999999	82.1407819302508
82.4	81.84568826031837
82.4	81.84568826031837
66.12	67.63742072961024
66.12	67.63742072961024
66.12	67.63742072961024
66.12	67.63742072961024
82.4	81.84568826031837
82.4	81.84568826031837
77.36000000000001	77.97698398562618
77.72	77.76282223062528
66.12	67.63742072961024
82.03999999999999	82.1407819302508
66.12	67.63742072961024
66.12	67.63742072961024
82.03999999999999	82.1407819302508

Figure 5. Actual vs Predicted WQI values using Linear Regression

6.2 Case-2: Water Quality Prediction using Logistic Regression

Logistic regression is used when the target/output variable is categorical, like good vs bad water quality. It predicts a probability of the output class. Even though, linear regression provides the WQI predictions, it still does not convey the potable status of the drinking water. Logistic regression can classify the water quality into categories like very poor, poor, good, excellent and unsuitable for a clear understanding of the problem. This is more suitable when final objective is a pass/fail classification for compliance reporting or alerts. In this experimental analysis, logistic regression returned a better classification accuracy of 99.34%, thus performing better than linear regression model. Figure 6 presents the categorisation of the predictions based on the water quality as per the model results.

```
Predicted: Very Poor Actual: Very Poor
Predicted: Very Poor Actual: Very Poor
Predicted: Very Poor Actual: Very Poor
Predicted: Very Poor Actual: Very Poor
Predicted: Very Poor Actual: Very Poor
Predicted: Very Poor Actual: Very Poor
Predicted: Poor Actual: Good
Predicted: Very Poor Actual: Very Poor
Predicted: Very Poor Actual: Very Poor
Predicted: Poor Actual: Poor
Predicted: Poor Actual: Poor
Predicted: Poor Actual: Poor
Predicted: Poor Actual: Poor
Predicted: Very Poor Actual: Very Poor
Predicted: Very Poor Actual: Very Poor
```

Figure 5. Actual vs Predicted Water Quality categories using Logistic Regression

From the results, it is inferred that most of the predictions of the water quality are precisely correct as seen from a sample of the predictions in Figure 5.

7. Conclusion

This research demonstrated a modern approach for large-scale water quality examination by integrating PySpark, machine learning models, and AWS cloud infrastructure. The study effectively utilized PySpark on SageMaker to handle the expansive NWMP dataset and train predictive models for water quality parameters addressing the research questions posed. The linear regression model achieved 97.38% accuracy in forecasting the Water Quality Index. The logistic regression model showed 99.34% accuracy in classifying drinking water as potable or non-potable. These results highlight the potential of the proposed approach in enabling scalable and efficient analysis of voluminous water quality data. This study establishes a foundation, but more investigation can go into more sophisticated machine learning methods such as XGBoost and neural networks to address non-linear models. Utilizing hyperparameter adjustment and cross-validation can enhance the performance of the model. Overall, this study showed that big data technology can extract useful insights from large water quality data sets. The proposed method can be improved in numerous ways to create sophisticated real-time water quality monitoring and management systems.

References

Albaldawi, W.S., Almuttairi, R.M. and Manaa, M.E., 2022, June. Big Data Analysis for Healthcare Application using Minhash and Machine Learning in Apache Spark Framework. In *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1-7). IEEE.

Alharbi, R., Alageel, N., Alsayil, M. and Alharbi, R., 2022. Prediction of Oil Production through Linear Regression Model and Big Data Tools. *International Journal of Advanced Computer Science and Applications*, *13*(12).

Berlian, M.H., Sahputra, T.E.R., Ardi, B.J.W., Dzatmika, L.W., Besari, A.R.A., Sudibyo, R.W. and Sukaridhoto, S., 2016, September. Design and implementation of smart environment monitoring and analytics in real-time system framework based on internet of underwater things and big data. In *2016 international electronics symposium (IES)* (pp. 403-408). IEEE.

Chiang, D.L., Wang, S.K., Wang, Y.Y., Lin, Y.N., Hsieh, T.Y., Yang, C.Y., Shen, V.R. and Ho, H.W., 2021. Modeling and analysis of Hadoop MapReduce systems for big data using Petri Nets. *Applied Artificial Intelligence*, *35*(1), pp.80-104.

Choumal, A. and Yadav, V.K., 2022, December. Evaluation of Fault Detection Algorithms for Photovoltaic Array Using Distributed Machine Learning Platform. In *2022 22nd National Power Systems Conference (NPSC)* (pp. 471-476). IEEE.

Hafez, M.M., Elfakharany, E.E.F., Abohany, A.A. and Thabet, M., 2023. Self-Tuning Parameters for Decision Tree Algorithm Based on Big Data Analytics. *CMC-COMPUTERS MATERIALS & CONTINUA*, *75*(1), pp.943-958.

Han, X., Shen, H., Hu, H. and Gao, J., 2022. Open Innovation Web-Based Platform for Evaluation of Water Quality Based on Big Data Analysis. *Sustainability*, *14*(14), p.8811.
Hemdan, E.E.D., Essa, Y.M., El-Sayed, A., Shouman, M. and Moustafa, A.N., 2021, July. Smart water quality analysis using IoT and big data analytics: a review. In *2021 International Conference on Electronic Engineering (ICEEM)* (pp. 1-5). IEEE.
Hu, L., Yang, S., Luo, X., Yuan, H., Sedraoui, K. and Zhou, M., 2021. A distributed framework for large-scale protein-protein interaction data analysis and prediction using mapreduce. *IEEE/CAA Journal of Automatica Sinica*, *9*(1), pp.160-172.
Kılınç, D., 2019. A spark-based big data analysis framework for real-time sentiment prediction on streaming data. *Software: Practice and Experience*, *49*(9), pp.1352-1364.
Kim, G.E., Steller, M. and Olson, S., 2020, September. Modeling watershed nutrient concentrations with AutoML. In *Proceedings of the 10th International Conference on Climate Informatics* (pp. 86-90).
Kimothi, S., Thapliyal, A., Akram, S.V., Singh, R., Gehlot, A., Mohamed, H.G., Anand, D., Ibrahim, M. and Noya, I.D., 2022. Big Data Analysis Framework for Water Quality Indicators with Assimilation of IoT and ML. *Electronics*, *11*(13), p.1927.
Kirschke, S., Avellán, T., Bärlund, I., Bogardi, J.J., Carvalho, L., Chapman, D., Dickens, C.W., Irvine, K., Lee, S., Mehner, T. and Warner, S., 2020. Capacity challenges in water quality monitoring: understanding the role of human development. *Environmental monitoring and assessment*, *192*, pp.1-16.
Krishnan, H., Pankajkumar, G., Poosari, A., Jayaraj, A., Thomas, C. and Joy, G.M., 2021, May. Machine learning based sentiment analysis of coronavirus disease related twitter data. In *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)* (pp. 459-464). IEEE.
Li, P. and Wu, J., 2019. Drinking water quality and public health. *Exposure and Health*, *11*(2), pp.73-79.
Ma, Y., Zhang, R., Cui, Z. and Lin, C., 2020. Projection based large scale high-dimensional data similarity join using MapReduce framework. *IEEE Access*, *8*, pp.121665-121677.
Mareeswari, V., Patil, S.S. and Ramanan, G., 2021. Real time sentiment analysis of Tweets using Apache Spark and Scala. *ACS Journal for Science and Engineering*, *1*(2), pp.9-15.
Meti, G. and G K, R.K., 2020, November. A Survey on Rainfall Analysis Using Big Data Analytics. In *Proceedings of the 2nd International Conference on IoT, Social, Mobile, Analytics & Cloud in Computational Vision & Bio-Engineering (ISMAC-CVB 2020)*.
Nair, J.P. and Vijaya, M.S., 2021, March. Predictive models for river water quality using machine learning and big data techniques-a Survey. In *2021 international conference on artificial intelligence and smart systems (ICAIS)* (pp. 1747-1753). IEEE.
Nie, X., Fan, T., Wang, B., Li, Z., Shankar, A. and Manickam, A., 2020. Big data analytics and IoT in operation safety management in under water management. *Computer Communications*, *154*, pp.188-196.

Ramani, R., Devi, K.V. and Soundar, K.R., 2020. MapReduce-based big data framework using modified artificial neural network classifier for diabetic chronic disease prediction. *Soft Comput.*, *24*(21), pp.16335-16345.
Ranganathan, D.G., 2020. Real time anomaly detection techniques using pyspark frame work. *Journal of Artificial Intelligence and Capsule Networks*, *2*(1), pp.20-30.
Scheili, A., Rodriguez, M.J. and Sadiq, R., 2015. Seasonal and spatial variations of source and drinking water quality in small municipal systems of two Canadian regions. *Science of the Total Environment*, *508*, pp.514-524.
Seal, A. and Mukherjee, A., 2019, April. Real time accident prediction and related congestion control using spark streaming in an AWS EMR cluster. In *2019 SoutheastCon* (pp. 1-7). IEEE.
Semic, A. and Karamehic, S., 2022. Stroke Analysis and Prediction Using PySpark, Suport Vector Machine and Random Forest Regression. *International Journal of Data Science*, *3*(2), pp.62-70.
Sirkiä, J., Laakso, T., Ahopelto, S., Ylijoki, O., Porras, J. and Vahala, R., 2017. Data utilization at finnish water and wastewater utilities: Current practices vs. state of the art. *Utilities Policy*, *45*, pp.69-75.
Vilas, M.P., Thorburn, P.J., Fielke, S., Webster, T., Mooij, M., Biggs, J.S., Zhang, Y.F., Adham, A., Davis, A., Dungan, B. and Butler, R., 2020. 1622WQ: A web-based application to increase farmer awareness of the impact of agriculture on water quality. *Environmental Modelling & Software*, *132*, p.104816.
Wen, Y., Li, M. and Ye, Y., 2020, April. MapReduce-based BP neural network classification of aquaculture water quality. In *2020 International Conference on Computer Information and Big Data Applications (CIBDA)* (pp. 132-135). IEEE.
Zhang, J., Sheng, Y., Chen, W., Lin, H., Sun, G. and Guo, P., 2021. Design and analysis of a water quality monitoring data service platform. *Comput. Mater. Continua*, *66*(01), pp.389-405.