Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

Jincheng Liu

A Synthetic Data-driven Solution for Urban Drinking Water Source Management

Master's thesis in Simulation and Visualization
Supervisor: Di Wu
June 2023

NTNU
Norwegian University of
Science and Technology

Jincheng Liu

A Synthetic Data-driven Solution for Urban Drinking Water Source Management

Master's thesis in Simulation and Visualization
Supervisor: Di Wu
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

NTNU
Norwegian University of
Science and Technology

ABSTRACT

Water quality monitoring plays a crucial role in urban water supply systems for the production of safe drinking water. However, the traditional approach to water monitoring in Norway relies on a periodic (weekly/biweekly/monthly) sampling of biological indicators, which fails to provide a timely response to changes in water quality. This thesis addresses this issue by proposing a data-driven solution that enhances the timeliness of water quality monitoring.

Our research team applied a case study in Ålesund Kommune. A sensor platform has been deployed at Lake Brusdalsvatnet, the water reservoir in Ålesund. This sensor module is capable of collecting data of 10 different physicochemical indicators. Leveraging this sensor platform, we developed a CNN-AutoEncoder-SOM solution to automatically monitor, process and evaluate water quality evolution.

There are three components in this solution. The first one focuses on anomaly detection. We employed a recurrence map to encode the temporal dynamics and sensor correlations, which were then fed into a Convolutional Neural Network (CNN) network for classification. It is noted that this network achieved an impressive accuracy of up to 99.6%. Once an anomaly is detected, the data is calibrated in the second component using various models including transformer, multiple layer perceptron, decision tree, and AutoEncoder. Since true values for calibration are unavailable, the results are evaluated through data analysis. Notably, the AutoEncoder, which applies a graph denoising principle, provides a calibration solution that closely matches the original clean data within time periods where water quality remains relatively stable. With high-quality calibrated data in hand, we proceeded to cluster the data into different categories to establish water quality standards in the third component. K-means served as the baseline clustering algorithm, while we also developed spectral clustering and Self Organizing Map (SOM).

The results revealed that SOM, utilizing AutoEncoder calibrated data, demonstrated the highest performance with a silhouette score of 0.73 which illustrates a small in-cluster distance and large intra-cluster distance when the water was clustered into three levels. This system not only achieved the objective of developing a comprehensive solution for continuous water quality monitoring but also offers the potential for integration with other Cyber Physical System (CPS) in urban management.

Additionally, this work is submitted to the Journal of Water Research.

Keywords: Water Quality Monitoring, Urban Water Supply System, Anomaly Detection, Signal Calibration

PREFACE

Time flies, and two years of life in NTNU are coming to an end. As I reflect on my journey as a graduate student, I am immensely grateful for the invaluable knowledge and experience I have gained. None of this would have been possible without the guidance of professors, the wholehearted assistance from my classmates and friends, and the unwavering support from my family. I sincerely thank all of you.

I am deeply appreciative of the time I have spent studying at NTNU. During this period, I had the opportunity to participate in various research projects. Through these experiences, I was able to apply the knowledge I acquired to real-world scenarios, gaining a fresh perspective on the subjects I had previously studied. These experiences transformed me from someone with limited background in information technology and programming into a master's student capable of addressing engineering problems. I am indebted to the support provided by the university.

Furthermore, I would like to express my profound gratitude to my supervisor, Prof. Di Wu. Under her meticulous guidance, this thesis was completed. From selecting the research topic, designing experimental plans, conducting theoretical analysis, and processing data, to writing and finalizing the thesis, she has offered me patient guidance and selfless assistance. In addition, she has provided generous help and technical support for my other work, deepening my understanding of data analysis and software development, and igniting my passion for data analysis. It is because of her support and mentorship that I chose this research topic.

I would also like to extend my gratitude to everyone who has contributed to this thesis. In particular, I would like to thank Razak Seidu, an expert in water treatment and the creator of the Vertical Profiler System, the sensor platform being used in this project. He not only provided valuable assistance in data processing and system validation for this thesis but also shared insightful background information, allowing me to develop a deeper understanding of the research context.

Lastly, I would like to express my appreciation to all the experts and professors who have taken the time to review and participate in the defense of this thesis. Thank you for your enlightening feedback! I hope that this is not just the end of my learning journey but the beginning of a new chapter.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ANN Artificial Neural Network. 9, 10

CNN Convolutional Neural Network. i, iv, 8, 9, 13, 15, 30, 56

CPS Cyber Physical System. i, 2, 7, 56, 57

CTS Correlated Time Series. iv, 3–5, 19, 20, 55, 56

LSTM Long Short Term Memory. 8

MLP Multilayer Perceptron. iv, 15, 18, 19, 32, 33, 35, 36, 38, 42, 47, 49

PCA Principle Component Analysis. 9, 10

ReLU Rectified Linear Unit. 19, 21

RNN Recurrent Neural Network. 8

RQ Research Question. 4, 5

SOM Self Organizing Map. i, iv, 10, 22, 24, 25, 43, 47, 49, 55, 56

SVM Support Vector Machine. 8–10

WSN Wireless Sensor Network. 8–10

CHAPTER

ONE

INTRODUCTION

This chapter aims to offer an overview of the project by presenting its motivation, research objectives, and research questions which establish the scope of the proposed research. Additionally, the chapter delves into a detailed discussion of the project's contributions, highlighting the insights and advancements it offers to the field of water quality monitoring. Lastly, an overview of the document's structure is provided, giving readers a clear roadmap of what to expect in subsequent chapters.

1.1 Background

The production of clean drinking water involves collaborative work that incorporates various physical, chemical, and biological treatments. Water treatment plants are designed to adapt their treatment processes based on the specific quality of the water [1]. However, surface water sources, such as rivers and lakes, which serve as the primary freshwater source, are susceptible to environmental changes and the introduction of pollutants from industrial and wastewater sources [2][3]. Therefore, monitoring the water quality in reservoirs is crucial from both ecological and economic perspectives. By assessing the quality of water in reservoirs in real time, the ecological impact can be evaluated, and appropriate measures can be taken to safeguard the availability of clean drinking water for the well-being of both ecosystems and human populations.

In the selection of indicators for water quality monitoring, there is no standardized requirement, and local governments can choose their own solutions. Both physicochemical and biological indicators are potential candidates [4]. Physicochemical parameters typically focus on intrinsic physical and chemical attributes of water, including temperature, pH, salinity, conductivity, dissolved oxygen, turbidity, and heavy metal content while biological indicators encompass the presence and abundance of bacteria, protozoa, viruses, and algae.

Traditionally, biological indicators have been more commonly used in water quality monitoring as they are believed to directly reflect water quality and provide valuable information. However, the collection of biological data is often labor-intensive, involving routine sampling of water and transporting them to a laboratory for analysis. This approach has drawbacks, including low sampling frequency, potential delays due to lengthy laboratory processes, and the possibility of

missing important events [5]. Additionally, monitoring biological indicators using sensors poses challenges. Although researchers have proposed solutions such as continuous monitoring of water quality by observing the behavior of rainbow trout [6], using biosensors for detecting urine in water bodies [7], or detecting particles with UV-Vis spectrophotometry [8], questions remain regarding their accuracy, reliability, and scalability.

Unlike biological indicators, continuous monitoring of physicochemical indicators offers the benefit of labor-free and real-time monitoring capabilities, facilitating more effective process control without or with fewer delays [9]. Recent research findings [10] suggest a mutual influence between physicochemical quality and biological indicators, indicating that monitoring physicochemical indicators could be a viable alternative. Moreover, there are now stable and accurate equipment options available for monitoring physicochemical indicators, ranging from portable devices [11] to large sensor stations [12], which can be customized to specific data collection requirements. This opens up possibilities for efficient data collection without the need for labor-intensive processes. Particularly, advancements in low-cost sensor technology have made monitoring such indicators more feasible. The cost and technological requirements of low-cost sensors have significantly reduced while maintaining stability and usability compared to wired sensors, enabling broader coverage and higher sampling frequency [13]. Leveraging these benefits, the use of physicochemical data can provide a more cost-effective and rapid-response approach to precisely measure water quality levels.

In this project, the water quality monitoring utilizes the Vertical Profiler System, a buoyed sensor platform developed by the Water and Environmental Engineering Group at NTNU in Ålesund, as depicted in Figure 1.1. The system incorporates various physical and chemical parameters to assess water quality, including temperature, conductivity, salinity, turbidity, pH, optical dissolved oxygen (ODO), and fluorescent dissolved organic matter (fDOM). The sensor platform is positioned at a specific location and collects data at different depths to capture variations in these indicators.

Figure 1.1: Actual picture of the YSI Vertical Profiler System sensor platform

The Vertical Profiler System can work as a sensor subsystem within a broader Cyber Physical System (CPS). A typical CPS architecture, as depicted in Figure 1.2, encompasses two domains and multiple subsystems that integrate various components such as sensors, computational units, controllers, and actuators through networking, enabling seamless interaction with physical objects and environments [14]. The primary objective of a CPS is to monitor real-time physical

behavior and achieve specific goals based on the acquired data [15] which is consistent with our target. Normally, within a complex CPS framework, the entities involved tend to exhibit a certain level of synchronization, leading to the generation of Correlated Time Series (CTS) during the process of collecting sensor data [16]. For example, in a traffic monitoring system, the traffic flow in different road segments can influence one another, resulting in a correlation among the collected time series, thereby forming a CTS dataset [17]. While the correlation among water quality sensors at different depths has not been extensively studied compared to traffic systems, the data collected in this project are regarded as a CTS dataset in some parts. This correlation is attributed to the spatial relationship that exists among the different time series within the dataset and the mutual influence from sensor to sensor. By acknowledging this correlation, it becomes possible to analyze the interdependencies and patterns present in the water quality data collected from multiple depths and sensors.

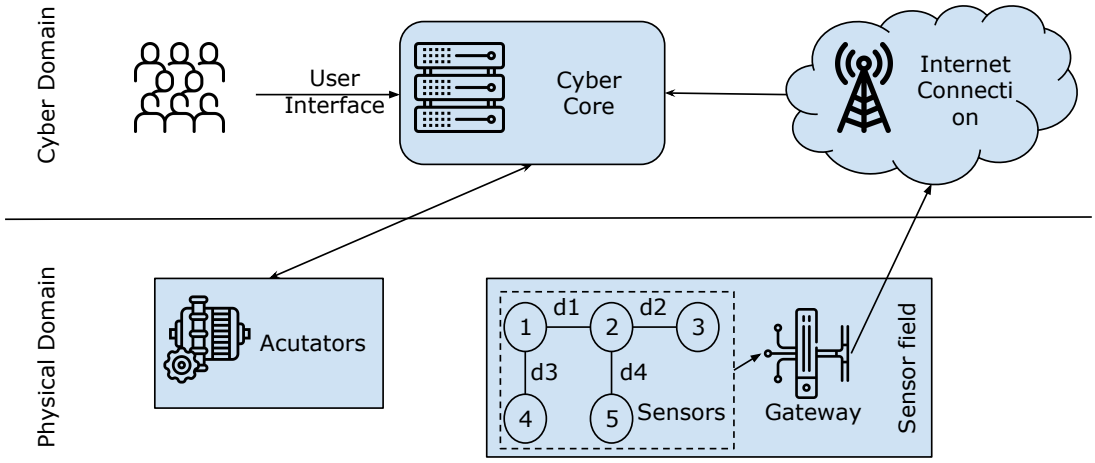


Figure 1.2: Traditional architecture of a Cyber-Physical System

1.2 Problem analysis

Currently, the Ålesund Kommune applied two standards for deciding the water quality. For the water source monitoring, current labor is put into the detection of heavy metal. However, according to the talk with the expert (Razal Seidu, professor in NTNU in Ålesund), it is not easy to separate different metals, and thus accurate measurement tends to be a problem. For processed drinking water, biological indicators are selected, as depicted in Figure 1.3. However, this approach presents challenges in achieving real-time monitoring. The current standard requires a minimum of three years of data which are collected biweekly to determine water quality, making real-time monitoring of water quality changes less feasible. This not only poses a significant challenge for environmental monitoring but also results in delayed responses from water treatment plants. Therefore, there is an urgent need for a new water quality system that can provide accurate and reliable information using data collected over shorter periods. Purely relying on biological indicators is insufficient to meet this need as explained in the previous analysis, prompting recent studies to focus on the monitoring of physicochemical data. However, three challenges arise when attempting to conduct data analysis in the context of water quality monitoring.

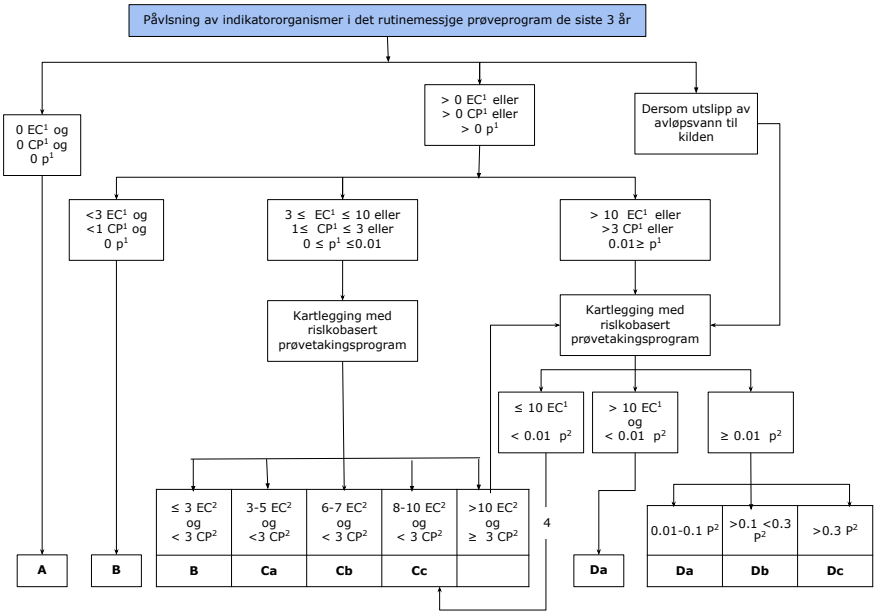


Figure 1.3: Current water quality level classification system. Adapted from [18]

The first challenge pertains to the quality of the collected data. Sensor platforms are typically sold as complete systems by manufacturers, making individual calibration for each sensor impractical [19]. Additionally, these sensors are often deployed in remote or inaccessible locations, as noted in several studies [20] [21], rendering traditional in-laboratory calibration methods less feasible. The absence of effective calibration procedures can result in inaccuracies in sensor data, posing challenges for conducting comprehensive data analysis. To address this issue, data-driven calibration techniques have emerged as a promising approach for obtaining more accurate datasets. Leveraging machine learning, a powerful tool for tackling similar challenges shows promise for exploration in this context.

The second challenge concerns the establishment of water quality standards. As mentioned earlier, different organizations and cities apply varying combinations of biological and physicochemical indicators for water quality monitoring. Therefore, a customized approach must be developed specifically for this occasion.

The third challenge arises from the previous consideration of the collected data as a CTS. In this regard, both the temporal dynamics and the synchrony among entities need to be taken into account. However, current studies in the water quality monitoring domain primarily focus on purely time series data, neglecting the investigation of synchrony among different indicators. Understanding the interdependencies and relationships among water quality indicators at various depths and locations can enhance our ability to identify patterns, detect anomalies, and gain a more holistic understanding of water quality variations.

Therefore, the objective of this research project is to devise an all-encompassing data-driven solution for the analysis of collected water quality data.

To achieve this objective, the following Research Question (RQ) are proposed:

- RQ1: What architectural design is required to optimize the collection, processing, and categorization of water quality data in urban water source management?

- RQ2: Which calibration techniques are effective in addressing data anomalies encountered during water quality monitoring?
- RQ3: How can suitable categories for assessing water quality in water sources be determined in the absence of prior knowledge?
- RQ4: What are the potential benefits and impacts of implementing this solution in Ålesund Kommune's water source management?

In response to these research questions, this project puts forth a data-driven solution and makes several significant contributions which can be summarized as follows:

- An innovative data-driven automatic system has been proposed for monitoring the water quality level in the Ålesund drinking water reservoir. This system incorporates advanced sensor technologies to continuously collect data and provide insights into the water quality parameters.
- The collected sensor data are analyzed and calibrated as CTS. This calibration process enhances the reliability and accuracy of the dataset, enabling more precise and meaningful analysis. By considering the interdependencies and relationships among different water quality indicators, the CTS analysis allows for a comprehensive understanding of the complex dynamics and variations in water quality over time.
- To further enhance data analysis and interpretation, the collected data are clustered into different levels using unsupervised machine learning algorithms. This clustering approach organizes the data into distinct groups or categories based on their similarities or patterns, enabling a more structured and systematic analysis.

Overall, through the proposed data-driven solution, the project aims to revolutionize water quality monitoring and analysis in the Ålesund drinking water reservoir. By combining advanced sensor technologies, CTS dataset, and graph-based clustering, this project strives to provide a comprehensive and reliable framework for drinking water quality monitoring.

1.3 Outline

This paper is structured as follows:

- Chapter 2 comprehensively reviews state-of-the-art methods in data-driven anomaly detection, water quality signal data calibration, and water quality clustering. The objective of this chapter is to establish the theoretical foundations for the proposed system by examining the latest advancements in relevant research. By reviewing the existing literature, this chapter aims to identify the key concepts, techniques, and methodologies that form the basis of the proposed system.
- Chapter 3 presents a detailed explanation of the system design and implementation specifics. This section outlines the technical details of the system, including three components used for its implementation.

- Chapter 4 focuses on the result analysis of the testing dataset and presents the results obtained from the case in Ålesund Kommune. This section aims to provide a thorough exploration and analysis of the results, including identifying anomalies, data calibration, and water quality clustering. By conducting a detailed examination of the results, this chapter contributes to a comprehensive understanding of the system's performance and highlights any areas that may require further investigation or improvement.

- Chapter 5 of the project report focuses on the validation and potential limitations of the project. The chapter provides an in-depth analysis of the expert evaluation conducted to determine the effectiveness and potential impact of the project. Additionally, the chapter discusses the factors that limit the performance of the architecture used in the project. Besides, additional contribution during the master's program is also listed.

- Chapter 6 draws conclusions based on the results obtained and discusses potential future work that can be undertaken to enhance the proposed architecture. This chapter summarizes the study's key findings, insights, and implications. Additionally, it explores avenues for future research and development, suggesting potential enhancements or extensions to the proposed system and addressing any limitations or challenges that have emerged during the study.

CHAPTER

TWO

RELATED WORK

In this chapter, a comprehensive review of relevant literature is presented. The chapter is divided into four parts based on the research objectives and questions. Section 2.1introduce the background information about urban water supply system and the usage of collected data in such system. Section 2.2 discusses signal classification techniques that aid in determining the appropriate method for detecting the anomaly. Subsequently, Section 2.3 explores various signal cleaning methods that can be employed once outliers are found. Finally, in Section 2.4, different methods for clustering water quality levels are explored.

2.1 Drinking water source monitoring

The urban water supply system plays a critical role in providing clean and safe drinking water to urban populations. It comprises various components, including water sources, distribution networks, and treatment and storage facilities [22]. In recent years, advancements in data collection technology have sparked research efforts aimed at improving the intelligence and security of the system, making it more resilient and efficient. For instance, Alaribi, Elazhari, and Zargelin proposed an automatic water supply system that enhances safety and reduces operational costs to meet the increasing demands of urban populations [23]. Similarly, Di, Wang, and Razak proposed a CPS architecture to meet sustainability requirements and improve the efficiency of the water supply system [24]. In both studies, data collection played a vital role, with sensor networks providing valuable insights into water quality dynamics in response to environmental changes. Effectively utilizing this data enables water utilities and authorities to make informed decisions, optimize system operations, and improve overall efficiency.

Other research focuses on the prediction of water quality. [25] used physical, chemical, and biological indicators to predict water source risks and provide decision-makers with suggestions. However, the inclusion of biological indicators in such predictions can result in slower processes due to the time-consuming nature of their assessment. [26] applied a deep neural network and deep matrix factorization to predict water quality by assessing the biological oxygen demand. This model considered only a single indicator, introducing uncertainty into the prediction process. [27] employed water images and an attentional neural network based on convolutional

neural networks to distinguish clean water from polluted water. Nonetheless, the accuracy of this network remains questionable, as polluted water may appear similar to clean water but still pose risks to human health.

These existing works serve as inspiration for the present project, which aims to analyze data collected from a sensor platform measuring ten physical and chemical parameters. By leveraging this data, the project seeks to develop a comprehensive solution for assessing water quality and addressing the challenges associated with existing prediction methods.

2.2 Anomaly detection

In this project, the Wireless Sensor Network (WSN) is utilized to collect time-series data from multiple sensors, enabling comprehensive data collection. To ensure data accuracy, advanced anomaly detection machine learning algorithms are applied to identify any intrinsic errors that may be present within the collected data. Numerous research efforts have been dedicated to addressing this challenge and developing effective solutions for accurate anomaly detection in time-series data obtained from WSNs. Currently, both unsupervised and supervised machine learning algorithms are employed for detecting anomalies in time-series data.

Unsupervised machine learning algorithms offer a versatile approach as they do not require labeled data and instead focus on capturing the underlying patterns within the dataset. Researchers can analyze the clusters generated by these algorithms and apply domain knowledge to interpret the meaning of each group. For instance, Munir, Siddiqui, Dengel, and Ahmed proposed a deep-learning-based unsupervised machine learning algorithm for anomaly detection, achieving an outstanding F1-score of 0.87, surpassing other algorithms at that time [28]. However, this method only considers single-parameter time-series signals, which may not fully utilize the multivariate dependencies often present in real-world WSN data. To address this limitation, Li et al. presented an algorithm based on Generative Adversarial Network with RNN-Long Short Term Memory (LSTM) layers to analyze the correlated data [29]. By testing their network on the dataset of real cyber physics system data, they proved the feasibility of their method. Besides, graph-based algorithms have emerged as another promising solution, leveraging the advancements in Convolutional Neural Network (CNN)s. For example, Zhang et al. designed a Multi-Scale Convolutional Recurrent Encoder-Decoder framework that processes multivariate time-series data using CNNs and Recurrent Neural Network (RNN)s, surpassing traditional algorithms such as CNNs, LSTM, and RNN [30]. Other research has also demonstrated the effectiveness of this approach, achieving F1-scores greater than 0.85 in test datasets [31] [32] [33].

On the other hand, supervised machine learning algorithms offer enhanced accuracy by leveraging labeled data. Through the analysis of a substantial amount of labeled data, the dataset can be thoroughly explored, enabling precise classification of out-of-sample data. For instance, Muriira, Zhao, and Min [34] employed Kernelized Linear SVM to establish spatial links among sensor data and identify anomalies. However, the increasing number of data parameters poses a challenge for most SVM-based anomaly detection algorithms, as the dimensionality becomes higher. To mitigate this issue, Borghesi et al. [35] utilized AutoEncoder to extract normal patterns and reduce the feature space, while Canizo, Triguero, Conde, and Onieva [36] applied one-dimensional CNN to extract features from individual sensors and classify them with RNN. Their studies

achieved high accuracy in industrial scenarios. Other supervised anomaly detection algorithms for WSN data, such as Principle Component Analysis (PCA) [37], ensemble learning [38], and deep learning algorithms like Temporal Fusion Transformer [39], have also shown promise in this domain. These studies not only validate the effectiveness of supervised machine learning in this field but also inspire the design of the current project, as discussed in Section 3.2.

In this project, the raw data initially lacked any labels. However, considering the effectiveness and accuracy of supervised machine learning, an interview was conducted with experts in the domain to obtain their assistance in labeling the data.

2.3 Water quality signal calibration

When it comes to sensor data calibration, machine learning has attracted more attention due to that it is critical to WSN deployment. As discussed in [40], there are three methods for low-cost wireless sensor calibration. The first and most traditional method is laboratory-based univariate linear regression, which involves calibrating individual sensors in laboratories. However, this method often overlooks factors such as temperature, humidity, and cross-sensitivity with other elements, which can result in inaccurate sensing in real-world scenarios. To address this limitation, empirical multivariate linear regression models have been proposed, which calibrate the sensor outputs with respect to target signals using predefined equations, after in-laboratory calibration. Another approach is the use of more advanced algorithms such as machine learning, which has also been explored in this project as well. Although machine learning requires training data, it offers greater adaptability to environmental changes and could deal with other complex regressions more efficiently.

Numerous supervised machine-learning algorithms have been studied in previous research. In a study by Guo et al. [41], the performance of Artificial Neural Network (ANN), random forest, and SVM regression were compared on a dataset collected from a small urban lake in northern China, with ANN showing the highest performance. However, Bao et al. [42] demonstrated that random forest also performed well on a different dataset while Tenjo et al. [43] obtained better results with SVM than ANN. In addition to these classic algorithms, CNN has also shown significant promise in this field as well. Maier, Keller, and Hinz [44] developed a highly accurate method for estimating chlorophyll concentration using one-dimensional CNN, which was proved to be able to be applied to real-world scenarios. Yu et al. [45] compared the performance of SVM regression and CNN for chlorophyll concentration, with CNN providing higher accuracy. Furthermore, researchers have explored combining different algorithms to improve performance. For example, Arnault et al. [46] combined ANNs with hierarchical agglomerative clustering, while Wang et al. [47] used a genetic algorithm-based SVM approach. Overall, the field of supervised machine learning for environmental applications is dynamic and diverse, with various algorithms showing promising results and researchers continually exploring new approaches and combinations to enhance performance.

Inspired by the findings from these studies, we decided to conduct our own experiments to test and compare the calibration efficiency for our chosen dataset. Details of our approach will be discussed in Section 3.3.

2.4 Water quality clustering

In the realm of water quality analysis, machine learning has gained significant traction. Given the high dimensionality of data collected from WSN, feature extraction plays a pivotal role in various applications in this field. Researchers have leveraged different techniques for this purpose, such as Self Organizing Map (SOM), a neural network-based clustering algorithm, which has been used for extracting lower-dimensional tensors to enable data visualization and pattern analysis[48][49]. PCA has also been explored. For instance, Salam, Salwan, Nadhir, and Riyadh utilized PCA to identify the essential parameters for constructing a water quality index in Iraq[50], while Mansi and Kumar employed it to shortlist the relevant water quality parameters for a feasible solution[51]. These studies showcase the potential of unsupervised learning in water quality classification. Notably, Xinguo et al. demonstrated that feature extraction could significantly enhance the performance of water quality prediction models, especially when dealing with imbalanced data[52].

Some researchers have focused on studying the trend of water quality, utilizing various machine learning techniques. For instance, Deng, Chau, and Duan employed ANNs and SVMs to predict seawater quality in Hong Kong based on a 30-year record of data[53]. Others have delved into understanding the relationship between specific water quality indicators and overall water quality levels. For example, Runzi, Jin-Hyun, and Ming-Han utilized Geographically Weighted Regression to identify the main factors contributing to declining water quality[54], while Yuanhong, Xiao, Zuoxi, Sunghwa, and Zong employed Lagoon to assess water quality based on reflectivity monitoring[55]. These projects have generated accurate results based on well-labeled data. However, these models may have limitations, such as being confined to specific areas due to the characteristics of the training data or the specialized sensors used. In other words, these models represent weak artificial intelligence solutions developed to address specific problems, whereas a more general solution is needed to achieve the objectives of this project.

On the other hand, there are also studies that have explored the use of unlabeled data in water quality analysis. Commonly used algorithms for water quality clustering include K-means, hierarchical clustering, and density-based spatial clustering of applications with noise[56]. For example, Fathi, Zamani, and Zare combined PCA and hierarchical clustering to group five sampling spots into three clusters[57]. Similarly, Mehdi, Yaser, Mohsen, Maryam, and Gengyuan utilized K-means to group water quality data from different locations[58]. In another study, Mohammadrezapour et al. employed both K-means and Fuzzy C-means to explore similarity from a spatial perspective[59]. Their researches showcase the potential of clustering algorithms for monitoring water quality.

Based on the analysis of previous work, this project tested different algorithms for clustering the sampling data and generating a standard to predict water quality based on the reservoir's physicochemical attributes with the help of unsupervised machine learning.

CHAPTER

THREE

METHOD FORMULATION

This chapter will primarily focus on the structure and deployment details of the data processing and analysis component, with a specific emphasis on the chosen algorithms. Additionally, an evaluation matrix will be introduced to estimate the performance of the system.

3.1 Overview of the solution

Figure 3.1 provides an overview of the proposed system, outlining its distinct components, each represented by a unique color. The foundation of the system lies in the data source, a sensor platform that supplies raw data for subsequent processing and analysis.

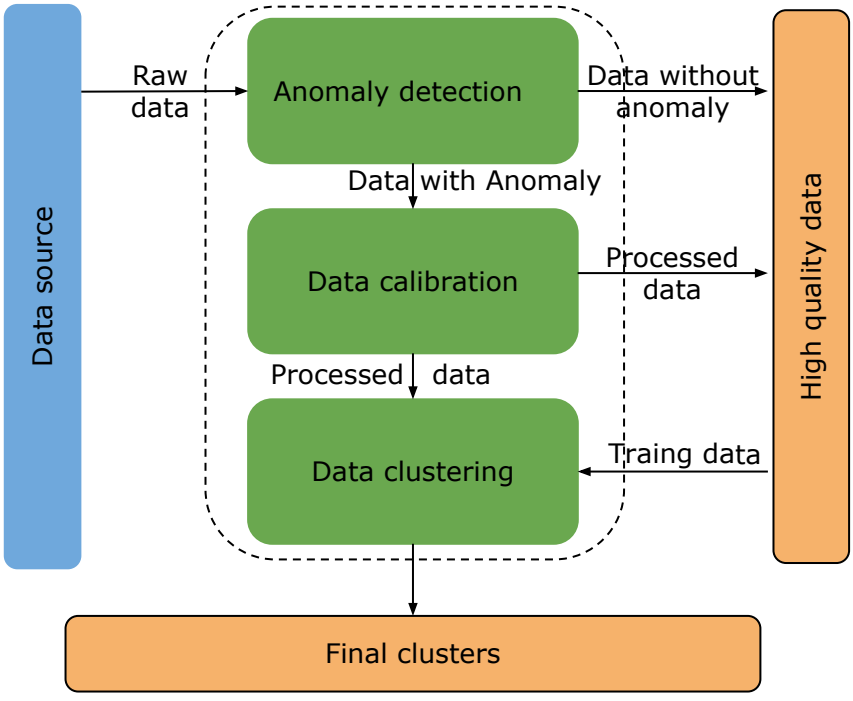


Figure 3.1: System overview of the whole project

The data processing and analysis component, indicated by the green color, plays a crucial role in the system's functionality. Initially, this component examines the incoming data to determine if calibration is necessary. If no calibration is required, the sensor data is designated as high-quality and is subsequently utilized in the water quality clustering process. Conversely, in the event that an anomaly is detected, the data is channeled into the signal calibration function. This calibration process ensures the production of calibrated data, which is then stored as high-quality data and fed into the water quality clustering model as well. The output of this model aids in generating timely water quality monitoring reports.

The final component, highlighted in orange, represents the ultimate output of the entire system. Once the data has undergone processing and analysis, the system yields two critical outcomes: high-quality data and water quality clusters. These high-quality data serve as valuable resources for further research, analysis, and decision-making in the field of water quality monitoring. The water quality clusters offer insights into the categorization and grouping of water quality data, enabling stakeholders to gain a comprehensive understanding of the overall water quality situation.

3.2 Water quality anomaly detection

The initial step after collecting raw data in this project is anomaly detection, which aims to classify the signal based on the presence of outliers and the type of fault. Figure 3.2 depicts the workflow. The data from various sensors located at the same place are combined using a recurrence map, as explained in Section 3.2.1. Subsequently, a CNN network is employed to classify the data into different categories based on the presence of anomalies, as explained in Section 3.2.2. The effectiveness of this anomaly detection component will be evaluated with training loss and accuracy and confusion matrix.

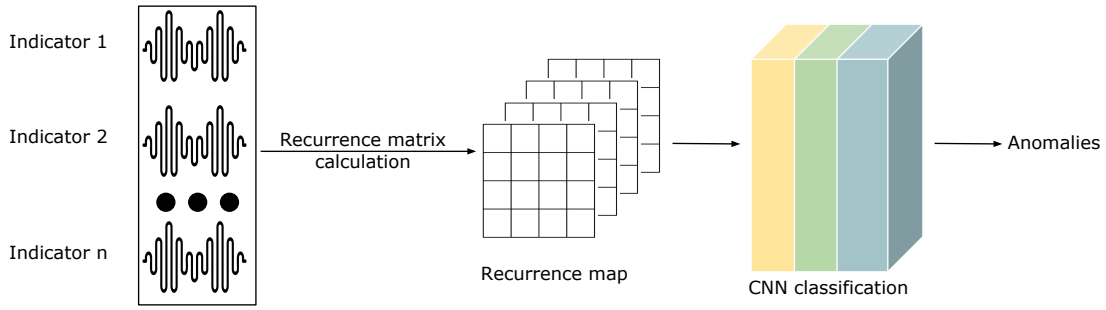


Figure 3.2: Processing for anomaly detection

3.2.1 Recurrence Map for data encoding

A recurrence map, as its name suggests, is a visualization tool used to analyze the repetition of data or patterns in given sequences. It works by comparing the distance between two states of a system, and expressing the difference using the following equation:

$$R_N(t,\tau) = \begin{cases} n & n\epsilon \leq \mid x(t) - x(\tau) \mid \leq (n+1)\epsilon \\ N & \mid x(t) - x(\tau) \mid > N\epsilon \end{cases} \tag{3.1}$$

In this equation, R represents the distance, while $x(t)$ and $x(\tau)$ denote the points in the sequence. ϵ is a predefined standard number used for measuring distance. If the distance between the two states is not greater than ϵ, R is calculated as the number of ϵ differences. Otherwise, it is calculated as the maximum difference, which is defined as N.

Algorithm 1 Recurrence matrix generation

```
function RecurrenceMatrixCalculation(s, eps, steps)
    if eps == None then
        eps ← 0.05
    end if
    if steps == None then
        steps ← 1
    end if
    d ← EuclideanDistanceCalculation(s)
    d ← floor(d/eps)
    d[d > steps] ← steps
    return d
end function
```

Once a signal was provided, the recurrence matrix will be calculated as the Pseudocode 1 denotes. Defaultly, the recurrence threshold, eps, is 0.05 and the steps are 1 if not defined. The Euclidean distance between every pair of statuses will be calculated based on Equation 3.1 and form the recurrency matrix which will later be transformed into the recurrence map plot.

3.2.2 CNN classifier design

This project employs a CNN for the purpose of anomaly detection and classification. The architecture of the CNN network is illustrated in Figure 3.3 and the input data is in the form of a 2D recurrence matrix generated from a recurrence map, with dimensions equal to a predefined window length. By leveraging convolutional and pooling layers, the input matrix is transformed into feature maps, which are further transformed into a feature vector. Lastly, a fully connected layer is applied to extract the classification of the sensor data. The functionality of the different CNN layers, namely the convolution layer, pooling layer, and fully connected layer, is depicted in Figure 3.4.

Convolutional layers (Figure 3.4.a) perform linear operations to transform the input from the previous layers into feature maps. The input data is divided into smaller elements, called receptive fields, with the same size as a predefined kernel or filter. At the same time, the same filter is applied to each receptive field through dot-product operations, allowing this layer to discover features from anywhere in the input matrix. One advantage is that the filters are not defined manually, but are learned through backpropagation, during which the parameters of the filters are modified based on the difference between predicted and real values. This adaptive nature of CNNs allows them to achieve high accuracy and adaptability to complex input data, as the learned filters can capture relevant features specific to the task at hand.

To address potential overfitting and enhance computational efficiency in CNNs, pooling layers (Figure 3.4.b) are employed. These layers introduce non-linearity and downsize the feature maps while retaining important features. The most commonly used pooling methods are max pooling and average pooling with a stride of 2 pixels. Max pooling selects the largest value within

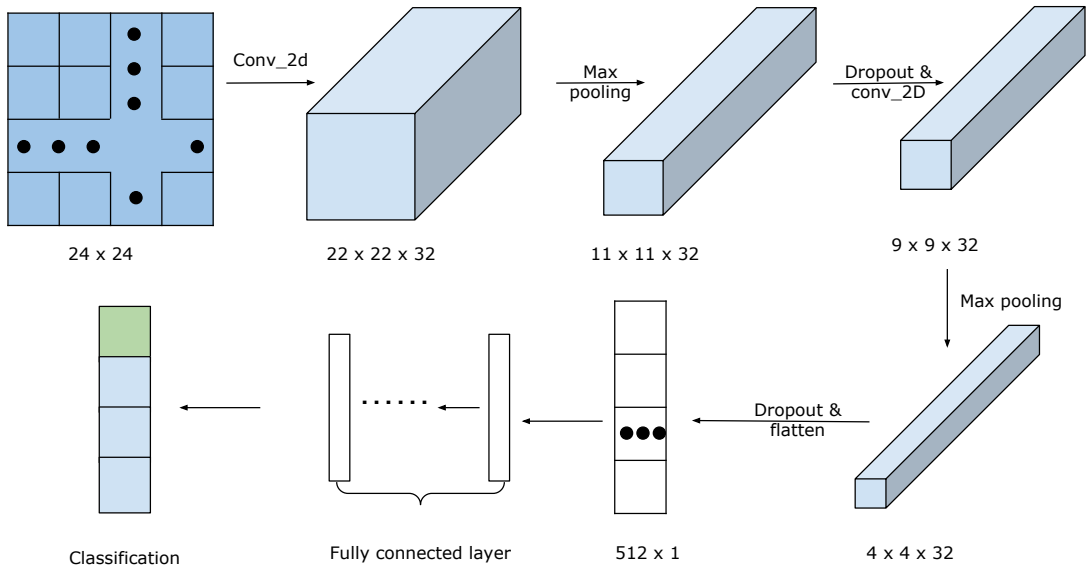


Figure 3.3: CNN network structure

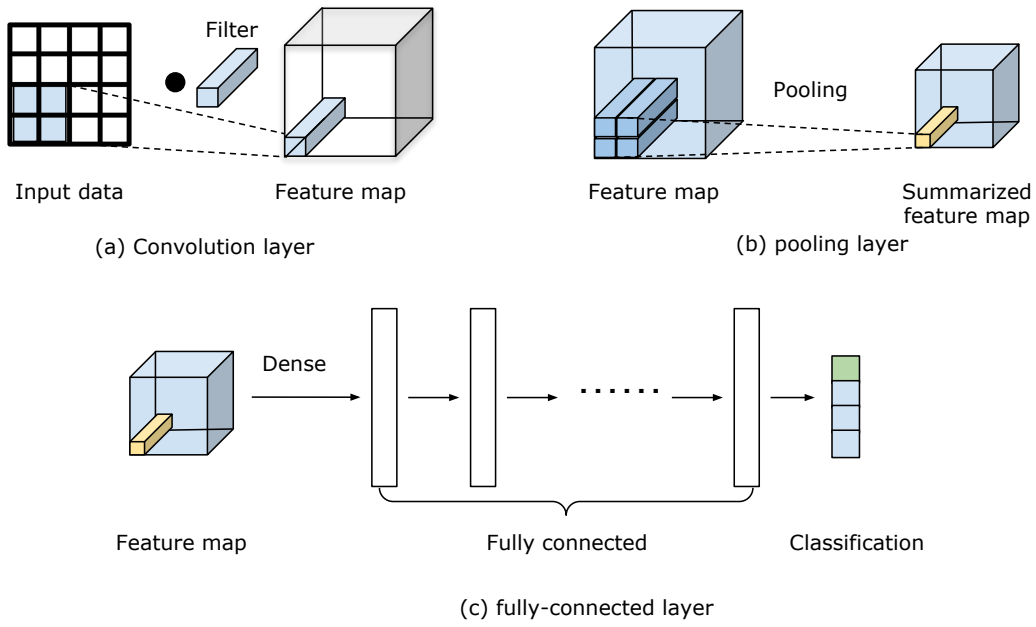


Figure 3.4: CNN layers working principle

each pooling window, while average pooling computes the average of the values. Compared to average pooling where the overall intensity is stored, Max pooling is preferred in our project as it captures the most significant features in local regions, allowing for the preservation of spatial information present in the recurrence map.

After generating the feature map, they will be transmitted to the fully connected layers(Figure 3.4.c). They are typically multilayer perceptron neural networks. By calculating the extracted features with

$$y = W_{FC}x \tag{3.2}$$

where x and y are the input and output respectively, W_{FC} is the linear weights. It is noted that Softmax activation function is used for the last layer of the CNN classifier since it is a multiclass classification problem. Softmax is an activation function that is used to normalize a vector such that the sum of all the elements in the resulting vector is equal to one. It is mathematically represented in Equation 3.3. Here, z_j represents the j^{th} element in a vector, and C represents the dimension of the input vector.

$$Softmax(z_j) = \frac{e^{z_j}}{\sum_{c=1}^{C} e^{z_c}} \tag{3.3}$$

By using this equation, the probability distribution among different classes can be calculated, and the trainable parameters can be adjusted to minimize the loss function which is the sum of the difference between the predicted output (weighted average) and the true output for each instance in the training data. For the proposed CNN classifier, the loss function is defined as Categorical Cross-Entropy, which is a commonly used loss function in multiclass classification. It is calculated using Equation 3.4, where y and $\hat{y}$ are the true one-hot encoded vector and prediction probabilities for each class of the i^{th} instance.

$$Loss = -\sum_{i=1}^{Output}{}^{size} y_i log(\hat{y}_i) \tag{3.4}$$

To reach a smaller loss, the model needs to generate a higher probability for the true class and lower probabilities for incorrect labels by adjusting the weights and biases in every layer and thus the model training can have an indicator.

3.3 Correlated time series calibration

After detecting anomalies in the sensor data, the signal calibration component is employed to correct the errors. As the data are correlated time series, both time dynamics and correlations can be considered in order to properly calibrate the data. If only time dynamics are considered, a Transformer model is utilized to analyze the past correct values from the same sensor and predict the true value for the anomaly. On the other hand, if only the correlation between different sensors is studied, MLP is applied to learn the relation between correct values from other sensors and anomalies. If both time dynamics and correlations are studied, the decision tree model and AutoEncoder model are chosen for the same purpose. Consequently, the calibrated

data can be generated, stored, and further analyzed.

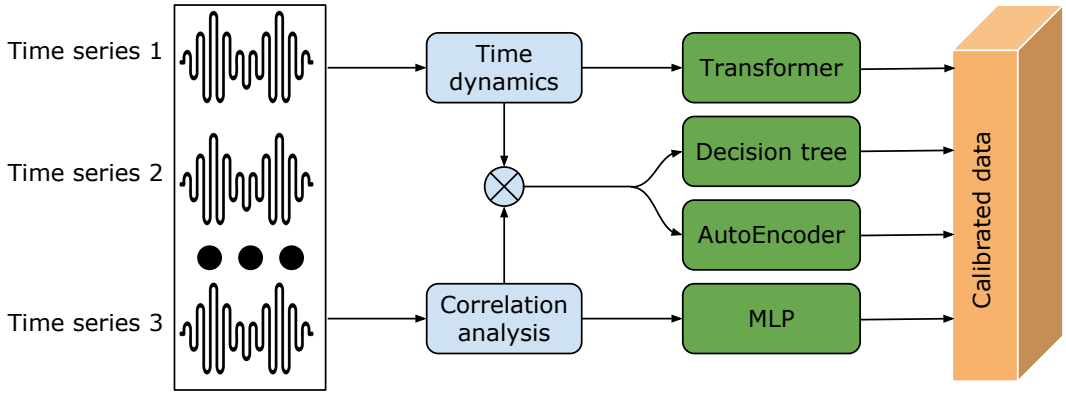


Figure 3.5: Workflow for the signal calibration

3.3.1 Time dynamics analysis with transformer network

The Transformer model proposed by Vaswani et al. in their influential paper [60] represents a significant breakthrough in the field of natural language processing. However, the innovative approach used in this model also has the potential to address the challenges of time series, which are relevant to the current project. In this work, we utilized a refactored version of the Transformer model to perform signal calibration, and Figure 3.6 illustrates the structure and principle of the applied model.

Unlike traditional machine learning models that depend on sequential alignment, the Transformer model employs a self-attention mechanism to process the time series data, as depicted in Figure 3.6.a. The input data is first read and embedded into a vector x_i, along with positional information, such as timestamps for time series analysis. By multiplying x_i with query weights W_Q, key weights W_K, and value weights W_V, a matrix of query vectors (Q, K, V), which is called an attention head, can be calculated. The elements of this matrix can be obtained through $q = x_i W_Q$, $k = x_i W_K$, $v = x_i W_V$, and the attention is calculated using the following equation if d_k is the dimension of the key vector:

$$Attention(Q, K, V)) = Softmax(\frac{QK_T}{\sqrt{d_k}})V \tag{3.5}$$

Where Softmax is the equation represented in Equation 3.3

In this project, we chose to use multi-head attention, which comprises multiple attention heads that can be calculated using the following equation:

$$MultiheadAttention(Q, J, K) = Concat(Attention(Q_i, K_i, V_i)) \tag{3.6}$$

In this equation, i represents the index of the attention head, and Q_i, K_i, and V_i are the relevant matrices for that specific attention head. W^O is the projection matrix for all the attention heads. The attention output is then added to the input vector of the layer and normalized to generate the results x_i^1 for further processing with a feed-forward network. Finally, the result is normalized into x_i^2 and fed into the next encoder layer or linear mapping layer. In other words, one encoder

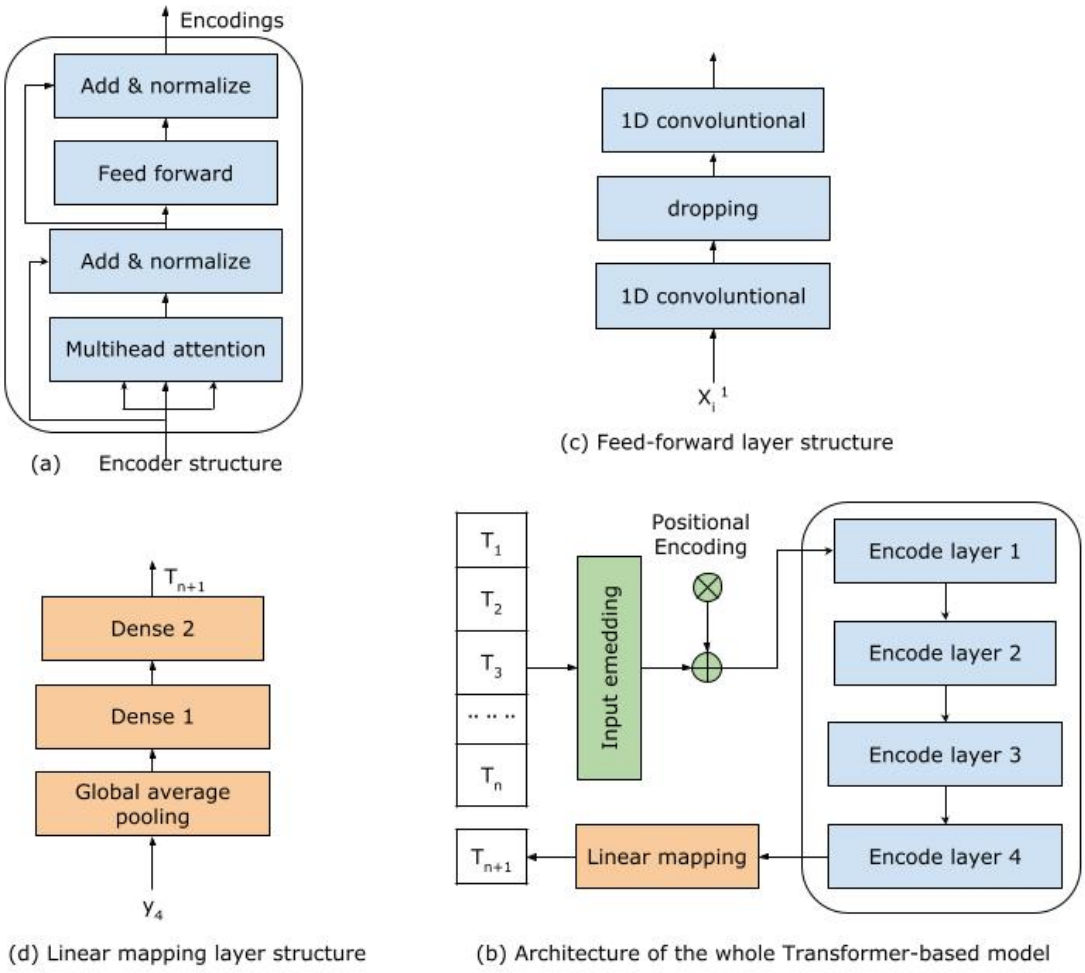


Figure 3.6: Transformer model architecture

layer with an input of x_i will conduct the following calculation [61]:

$$x_i^1 = Norm(x_i + MultiheadAttention(Q, J, K)) \tag{3.7}$$

$$x_i^2 = Norm(x_i^1 + FeedForward(x_i^1)) \tag{3.8}$$

In the encoder layer of the feed-forward network, two 1D convolutional layers were utilized to extract the relevant features. The structure of this layer is illustrated in Figure 3.6.b., which includes a dropout layer to minimize overfitting, two convolutional layers, and a feature map that encompasses essential information from x_i^1 was calculated.

Finally, a MLP layer generates the ultimate result. Following the average pooling process that reduces the dimensionality, the resultant vector is directed towards the dense layers, as shown in Figure 3.6.c. This approach enhances the model's ability to make time-series predictions.

Therefore, the architecture of the Transformer model is presented in Figure 3.6.d. The input of this model is a time series with a predefined window length, denoted by n. The input sequence is then processed by the input encoder along with the position encoder, followed by four encoder layers. Finally, the output is passed through linear mapping functions to predict the next value in the time series.

3.3.2 Correlation analysis with Multilayer Perceptron model

The MLP is a neural network that can enhance the mapping capability of input data. Figure 3.7 illustrates the architecture of MLP model which is applied in this project. It is composed of three layers: input, hidden, and output, and the calculation nodes are called perceptrons. In this approach, the values are initially fed into the input layer of the MLP. Then, computations occur through multiple hidden layers, leading to the generation of calibrated values from the output layer.

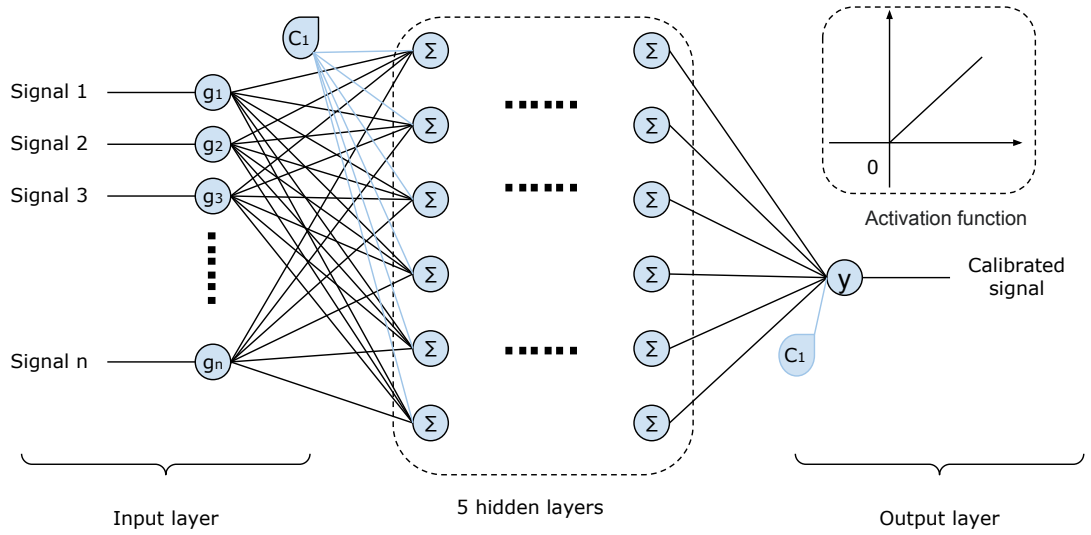


Figure 3.7: MLP neural network architecture

If we represent the inputs and computation results of each perceptron by $[I_1, I_2, \ldots, I_{h_1}]$ and

[O_1,O_2, ...,O_{h_2}] where h represents the number of input perceptrons, the output can be calculated using Equation 3.9.

$$O_j = \sum_{i=1}^{h} W_{ij} I_i + c_j \tag{3.9}$$

In this equation, j is the number of perceptrons. w_{ij} represents the link weight between two nodes, and c_j is the bias value. However, these results require further processing with the activation function. Among various activation functions, the Rectified Linear Unit (ReLU) is the most commonly used due to its better data representation ability compared to other activation functions, such as sigmoid and hyperbolic tangent [62]. The final output of a perceptron can be represented as shown in Equation 3.10.

$$f_j(x) = \mathsf{max}(0, O_j) = \begin{cases} 0 & \text{if } O_j \leq 0 \\ O_j & \text{if } O_j > 0 \end{cases} \tag{3.10}$$

This activation function is applied to all the perceptrons in this project, and thus the final results can be obtained from the output of the last hidden layer, as shown in Equation 3.11.

$$y_i = f_j(\sum_{i=1}^{h} W_{i,j} I_i + c_j) \tag{3.11}$$

In this project, the input to the MLP network is the sensor values that have been deemed accurate based on anomaly detection, while the output is the calibrated sensor values. By training the MLP network to learn the relationships among different sensors at various locations, the model can analyze the entity correlations among them. Therefore, when calibrations are needed, the predicted values can replace the measured values.

3.3.3 Correlated Time Series analysis with decision tree algorithm

As a non-parametric supervised machine learning algorithm, the decision tree is a widely used hierarchical tree structure in both classification and regression tasks. Figure 3.8 illustrates the basic principle of a decision tree.

The tree is composed of three types of nodes, starting from the root node that receives input data and passing through various internal nodes that receive data from the previous layer. The feature space is then partitioned based on a split θ, where the feature j and threshold t_m are defined As the following equation illustrates:

$$Q_m^{left}(\theta) = (x, y)|x_j \leq t_m Q_m^{right}(\theta) = Q_m \; Q_m^{left}(\theta) \tag{3.12}$$

The input data at node m is denoted by Q_m, where each data point consists of a feature vector x and a target value y. Using θ, the feature space is divided into two disjoint parts. Based on the comparison between x_j and t_m the left feature space $Q_m^{left}(\theta)$ and the right feature space $Q_m^{right}(\theta)$ group similar samples. The quality of a split is measured by the weighted average of the loss function H as follows:

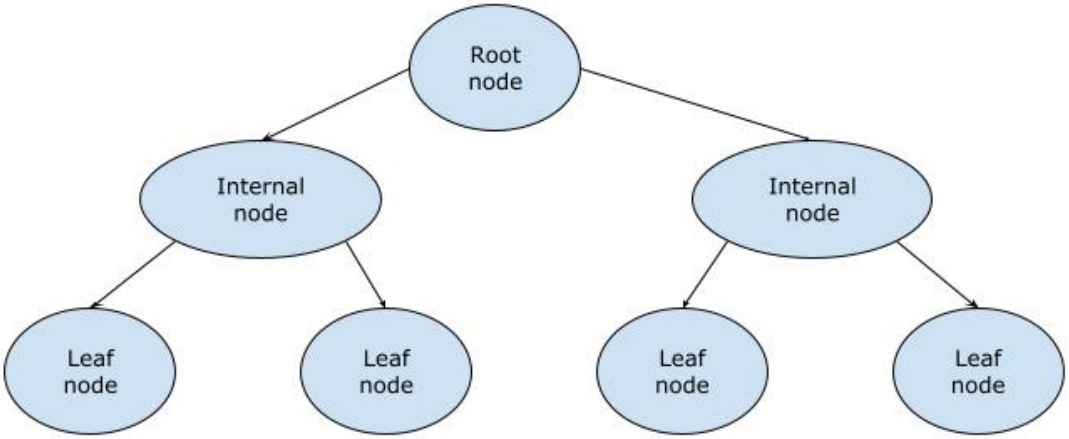


Figure 3.8: Decision tree structure overview

$$G(Q_m, \theta) = \frac{n_m^{left}}{n_m} H(Q_m^{left}(\theta)) + \frac{n_m^{right}}{n_m} H(Q_m^{right}(\theta)) G(\theta_m, \theta) \tag{3.13}$$

Here, n_m represents the total number of samples at node m, while n_m^{left} and n_m^{right} indicate the number of samples in the left and right feature spaces, respectively. The optimal split is the one that minimizes $G(Q_m, \theta)$. This decision process is repeated until the tree reaches its maximum depth or pure leaf nodes where $n = 1$ are reached. The possible results are then represented as leaf nodes in the tree.

The project employs a decision tree to calibrate signals by having it learn the correlation between input features - such as spatial information and precise sensor data - and the sensor data requiring calibration.

3.3.4 Correlated Time Series analysis with proposed AutoEncoder-based algorithm

AutoEncoder is a type of neural network trained to generate an output identical to the input. Although x'_j and x_j would not be exactly the same, they keep in consistency to each other in probabilistic terms such as mean and standard deviation [63]. And thus it is commonly utilized for dimensionality reduction, image denoising, and anomaly detection. The AutoEncoder model can be divided into two parts: the encoder, which receives input data and processes it to generate a code, and the decoder, which regenerates the data from the code to make it as similar to the input data as possible. This process enables the AutoEncoder to extract important features from the input data and remove noise. As shown in Figure 3.1, the proposed AutoEncoder model will take both temporal dynamics and correlation into consideration. The detailed model is shown in Figure 3.9.

At the beginning of this model, data are encoded into a 2D tensor where both temporal and correlation information will be considered. The tensor is of dimension $10 \times 20 \times 1$, where 10 indicates the number of indicators considered, 20 is the length of the time series window and 1 is the channel number. Specifically, each tensor is comprised of 20 samples containing all the measured parameters. The input data is processed using the pseudocode in Algorithm 2 to generate the 2D tensor $x_i \epsilon R^{10\times20\times1}$ once the input data and time series window length are

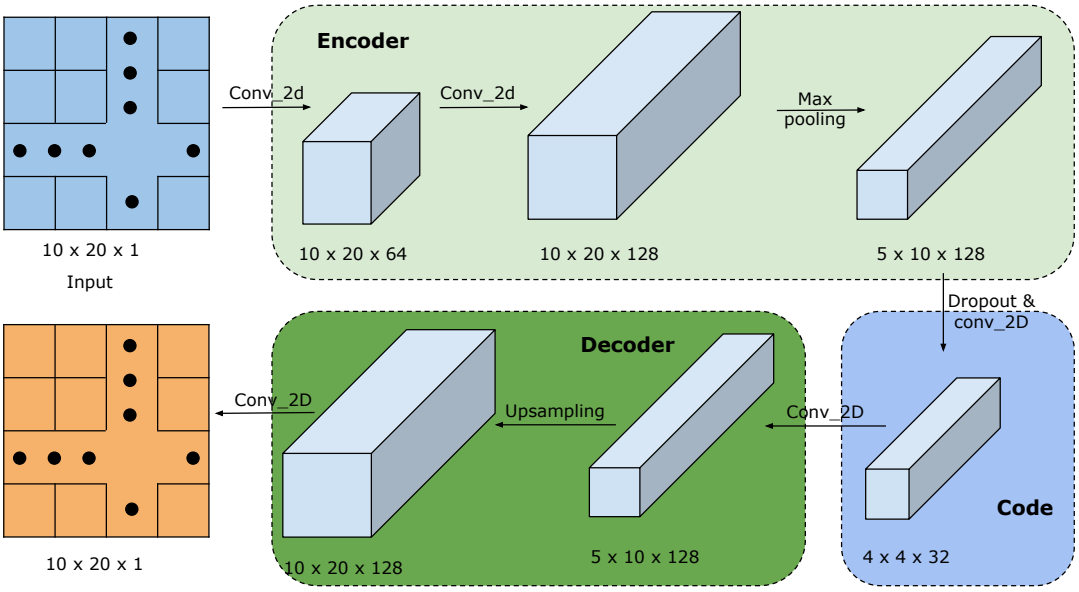


Figure 3.9: AutoEncoder structure overview

decided.

Algorithm 2 Encode data for AutoEncoder calibration

```
function CreateTensorForAutoEncoder(inputdata, window, parameters)
    initialize samplelist and sample
    depths ← unique depths in inputdata
    for each depth in depths do
        depthdf ← inputdata[depth == d]
        for each i from 0 to length of depthdf − window do
            segment ← vector obtained by reshaping depthdf[i : i + window].values
            sample.append(segment)
        end for
        samplelist.append(sample)
    end for
    return samplelist
end function
```

The AutoEncoder operation can be represented by an encoder operation (f) and a decoder operation (f') and the trainable parameters can be represented by $\theta = W, W', b, b'$. The compressed code can be expressed as

$$h^j = s(x_i * W_j + b_j) \tag{3.14}$$

Where $W = w_j, j = 1, 2, ..., n$ and $b = b_j, j = 1, 2, ..., n$ are the weight matrix and bias vector for the encoder layer, $*$ is the convolution layer operations and s is the activation function. In this study, the ReLU activation function, as shown in Equation 3.10, was selected for the encoder and decoder.

After calculating the code, if $W' = w'_j, j = 1, 2, ..., n$ and $b' = b'_j, j = 1, 2, ..., n$ are the weight matrix and bias vector for the decoder layer, the regenerated data can be obtained through:

$$x'_j = s(\sum_{j \in H h^j * W'^j + c}) \tag{3.15}$$

Where c is the bias and H is the collection of feature maps.

Unlike the traditional AutoEncoder model, the error is considered as the difference between uncorrupted data and regenerated data while estimating the effectiveness of the proposed model. In this project, the sigmoid function, as shown in Equation 3.16, was chosen to measure the cross-entropy error and thus minimize it during the training process.

$$Sigmoid(x) == \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x} \tag{3.16}$$

3.4 Water quality clustering

In this section, we utilized SOM and spectral clustering algorithms to cluster the water quality data, which was first calibrated and processed to generate high-quality signal data through the methods mentioned in the last section. In order to assess the effectiveness of these algorithms, we used the K-means algorithm as a baseline and evaluated their performance by employing a combination of evaluation metrics, namely the silhouette score and the distortion. Furthermore, these metrics were used to determine the optimal values of critical parameters for the algorithms, such as the number of clusters for K-means, sigma for SOM, and affinity for spectral clustering.

The principles and designs of the clustering algorithms, as well as their evaluation methodologies, are presented in this section.

3.4.1 K-means algorithm

The k-means clustering algorithm is a technique for grouping data points based on their centroids. Despite being an NP-hard problem, it is possible to compute a minimal local solution. Initially assigned to a random class, the algorithm aims to determine a set of centroids that minimize within-cluster variance. To achieve this, the algorithm calculates the distance between each data point in X and its closest centroid using the function:

$$dis = \sum_{x \in X} [f(C, x) - x]^2$$

Here, C is the center of every cluster, and for every x in the data set, f(C,x) is the function to find the nearest centroid to x. The pseudocode is shown as Algorithm 3 shows. Once the number of clusters is defined, the training process will start with random centroids, and then the clustering will be made based on the distance to every centroid where the minimum distance represents the belonging. In every iteration, they will be updated to the mean value of all the data points in the cluasters until convergence. The worst complexity is $O(m^{\frac{K+2}{p}})$ where K is the number of clustering and p is the number of features [64].

Algorithm 3 K-means algorithm

Require: Set of data points $\mathcal{X} = x_1, x_2, \ldots, x_n$, number of clusters k
Ensure: Set of cluster centers $c_1, c_2, \ldots, c_k$
  Initialize k cluster centers randomly
  while cluster centers have not converged do
    for each data point $x_i \in \mathcal{X}$ do
      Assign x_i to the nearest cluster center c_j
    end for
    for each cluster center c_j do
      Update c_j to be the mean of all data points assigned to it
    end for
  end while

3.4.2 Spectural clustering

The concept of spectral clustering was introduced by Andrew Ng, Michael Jordan, and Yair Weiss in their seminal paper [65]. It examines the similarities among data points to perform clustering. However, it incorporates elements from graph theory, linear algebra, and optimization to consider the global structure of the dataset. Unlike traditional K-means clustering, spectral clustering leverages the information contained in the smallest eigenvalues of the Laplacian matrix to extract the underlying structure of the original dataset. Consequently, it has found wide applications in data mining and image clustering.

To implement spectral clustering, the raw high-quality data is first encoded, resulting in the generation of images. Subsequently, the clustering process follows the pseudocode outlined in Algorithm 4. The affinity matrix W is initially computed to analyze the similarities among data points. The calculation of W_{ij}, the element at the i-th row and j-th column of W, is performed using the equation:

$$W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} \tag{3.17}$$

The distance between data points i and j, denoted as $\| x_i - x_j \|$, represents the dissimilarity between them. Once the affinity matrix W is computed, the Laplacian matrix L can be generated using the equation:

$$L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}} \tag{3.18}$$

where D is a diagonal matrix and its elements are computed as:

$$D_{ii} = \sum_j W_{ij} \tag{3.19}$$

It is noted that L captures the relationship between data points by considering both the connectivity and the degree of each data point. By calculating the K (desired number of clusters) smallest eigenvalues and eigenvectors, spectral clustering obtains a low-dimensional representation of the data.

Algorithm 4 Spectral Clustering

Require: Set of $InputData$ points $\mathcal{X} = x_1, x_2, \ldots, x_n$, number of clusters k

Ensure: Set of $ClusterCenters$ $c_1, c_2, \ldots, c_k$

$W \leftarrow$ affinity matrix from $InputData$

$L \leftarrow$ normalized Laplacian matrix from W

$X \leftarrow$ the first K eigenvectors corresponding to the smallest eigenvalues of L

$ClusterCenters \leftarrow$Apply K-means to the rows of X

3.4.3 Self Organizing Map

SOM clustering is an unsupervised machine-learning algorithm. It is first proposed by Teuvo Kohonen in 1982 in their paper [66]. Basically, it receives n-dimensional input vectors and feeds them into a neuron network to generate a two-dimensional map that could retain the original information in the input dataset. This map preserved the structural information of the data points in the data set which, in other words, provides similar interconnecting weights to the neighboring points. Moreover, the map itself also contains information about centroids. Every point on the map is related to the interconnecting weights and the value of it represents the centroids. In this project, a SOM-based model was proposed to cluster the water quality data. The topology is depicted in Figure 3.10.

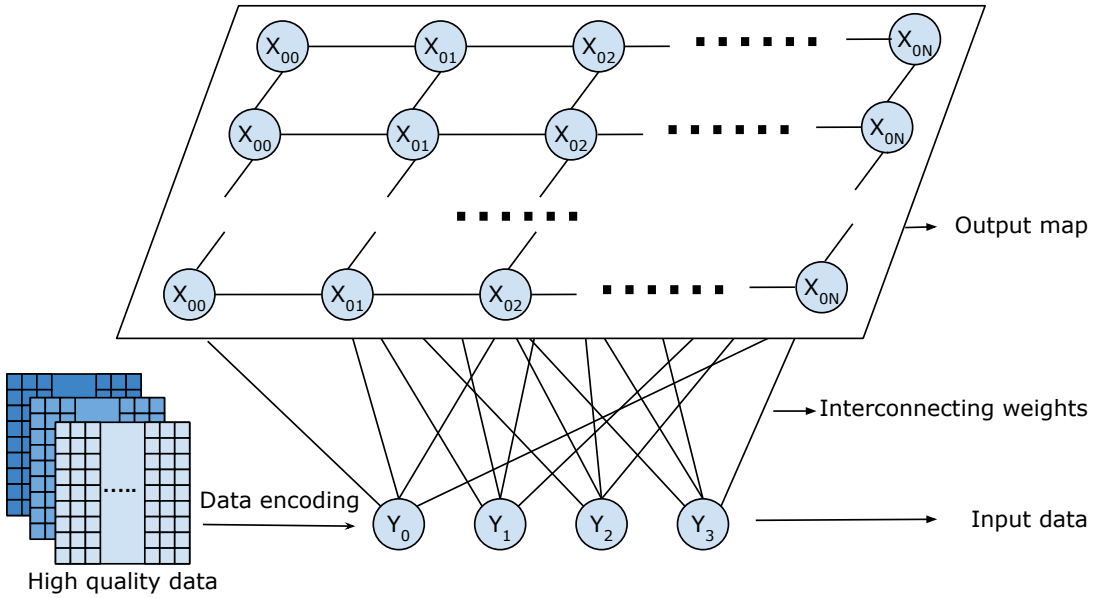


Figure 3.10: Topology of self-organizing map

The present model utilizes input data derived from the high-quality data obtained in the previous chapter. To comply with the requirements of the sensor platform, the data is structured as a 3D tensor with the dimensions of location, sensors, and timestamp. The first step involves encoding this tensor into a 2D image, which organizes the data collected at the same time. One dimension of the image represents the depth, while the other dimension comprises different sensor data. This transformation changes the problem into an image clustering task while preserving the spatial correlation among the data. Moreover, by analyzing the data collected at various timestamps, the temporal changes in water quality can be investigated in detail. Then the images are processed through a SOM algorithm, which analyzes the input data at the pixel level

to generate a Kohonen map. The training process for SOM is outlined in Algorithm 5.

Algorithm 5 Self-Organizing Map Algorithm

Require: Set of $InputData$ points $\mathcal{X} = x_1, x_2, \ldots, x_n$, stopping criterion
Ensure: Set of $ClusterLabels$ $label_1, label_2, \ldots, label_k$
  Initialize the weights of the nodes randomly
  repeat
    Select a data point randomly from the input dataset
    $distance \leftarrow$ the Euclidean distance between the selected point to all other points
    $BMU \leftarrow$ the point with closest weight vector
    Update the $weights$ of BMU and its neighboring nodes
  until stopping criterion is met

Initialization of weight values is done randomly for each input sample, and then the weight at the best matching unit and its neighboring weights are updated with a learning rate. The best matching unit is defined as where the distance between a sample and other weighted vectors is minimized and thus can be computed as

$$\parallel x - m_c = min \parallel x - m_i \tag{3.20}$$

where x is the sample vector and m_c and m_i are the best matching unit and i^th weighted vector respectively. The neighboring weighted vectors at $t + 1$ can be then calculated based on the value at time t as

$$m_i(t+1) = m_i(t) + \eta(t)h_{ci}(t)(x - m_i(t)) \tag{3.21}$$

Here $\eta(t)$ is the learning rate which would decay with time following

$$\eta(t) = \eta(0)exp(-\frac{t}{\tau_1}) \tag{3.22}$$

and h_{ci} denotes the neighboring kernel and is expressed as

$$h_{ci}(t) = exp(-\frac{d_{ci}^2}{2\sigma^2(t)}) \tag{3.23}$$

$$where \quad \sigma(t) = \sigma(0)exp(-\frac{t}{\tau_2}) \tag{3.24}$$

Here, d_{ci} denotes the distance from the best matching unit to the I^{th} neighboring data point rate while $\sigma(t)$ is the radius of the neighboring area which follows an exponentially decaying function.

The updating will finish when the convergence requirement or iteration number is reached. Normally, the convergence would be the dissimilarity, which can be calculated as the distance between data points. This project achieves this by using the Euclidean distance as in K-means. However, unlike K-means, which directly obtains the centroid, SOM uses weighted vectors to approximate the centroid. In other words, the weighted vectors play a crucial role in future prediction.

3.5 Evaluation matrix

In this project, the key parameter that needs to be defined is the number of clusters, denoted by K. The selection of K is performed by comparing the combined results from the distortion which are used to evaluate the tightness and the silhouette score which is used to evaluate the clustering performance based on cohesion and separation.

Distortion, also referred to as intra-cluster distance or within-cluster sum of squares, is a widely employed measure for assessing the performance of clustering algorithms. It quantifies the average squared Euclidean distance between each data point and its corresponding cluster centroid. By evaluating distortion, researchers can determine the quality of clustering results and make informed decisions regarding the optimal number of clusters.

For the silhouettes score, cohesion refers to the distance from a data point to its cluster's centroid, while separation refers to the distance from this point to other clusters' centroids. Specifically, the silhouette score measures the ratio of cohesion to separation, which is calculated using the following equation:

$$S_i = \frac{b_i - a_i}{\mathsf{max}(a_i, b_i)} \tag{3.25}$$

Here, b_i is the mean distance from the i^{th} data point to all other data points in the same cluster, and a_i is the average intra-cluster distance from the i^{th} data point to all other clusters' centroids.

Silhouettes Score has a range from -1 to 1, where a higher index indicates higher inter-cluster similarity and lower intra-cluster similarity. Specifically, 1, 0, and -1 denote the best, indifferent, and wrong clustering results, respectively.

However, both matrices have their own limitations. The distortion does not take the intra-cluster similarity into consideration and the silhouette score is sensitive to noise and cannot handle overlapping clusters. Therefore, the combined results of both metrics are used to determine the final number of clusters.

CHAPTER

FOUR

CASE STUDY IN ÅLESUND KOMMUNE

In this section, we present the outcomes obtained from the system using test data. In addition to detailing the data source, we evaluate the results using various evaluation metrics. For anomaly detection, we employ accuracy measures and a confusion matrix. To analyze the distribution of the calibrated data, we utilize data analysis techniques. Furthermore, we assess the final clustering results using silhouette scores and distortion measures. To support transparency and reproducibility, the source code for the implemented system is available in the referenced source [67]

4.1 Data collection

As shown in Figure 4.1, the whole sensor platform is located at the center of Brusdalsvatnet Lake, the drinking water reservoir in Ålesund, Norway, whose position is 62.48 degrees north and 6.47 degrees east. Every 12 hours, the platform will measure the physical attributes, including temperature, conductivity, salinity, turbidity, and PH, and chemical parameters, including optical dissolved oxygen(ODO) and Fluorescent, dissolved organic matter (fDOM). Together with these data, the timestamp of sampling time and the depth will be packaged and then transferred to a local server via a 900 MHz radio connection. All the data are stored in the form of correlated time series where the temporal information is represented by timestamp and the entity correlation can be among different sensors and different spatial locations.

Exploratory data analysis is an essential step in gaining insight into the properties of historical data, especially in understanding the distribution and patterns of each parameter. The dataset utilized in this project covers the period from June 9th, 2020, the project's starting date, to August 19th, 2022. The flattened and summarized training dataset is presented in Table 4.1, where the Count column indicates the number of observations for each parameter, while the Min and Max columns indicate the minimum and maximum values for each parameter. The Mean column presents the average value for each parameter, and the Miss data column shows the number of missing values for each parameter. Although only temperature and fDOM OSU have a small portion of missing data, the total amount is less than anticipated due to the platform's inoperability during the lake's freezing periods or the sensor platform's maintenance.

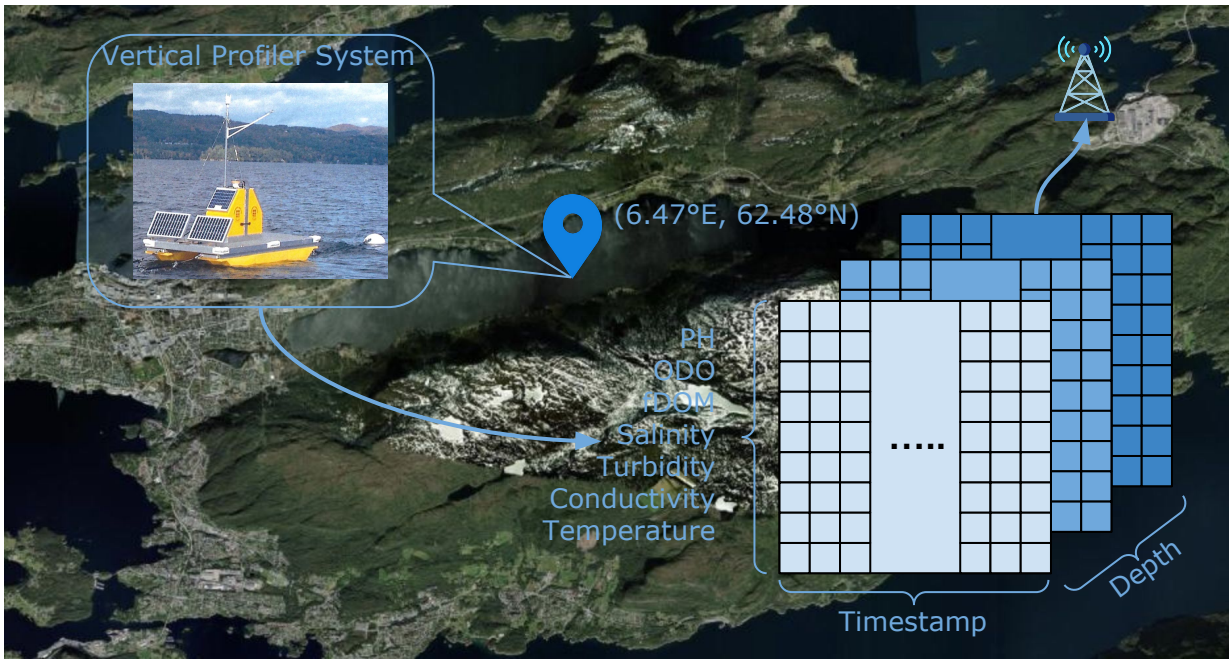


Figure 4.1: Information about the sensor platform

By examining the mean values, it is apparent that not all the data are reliable. For instance, the average PH should not be 2.60, which indicates a strong acid. Moreover, considering both the minimum and maximum values, it is apparent that the raw data contains outliers in all indicators except timestamps and depths.

To investigate the events that occurred on the platform, Figure 4.2 illustrates the temporal dynamics of each sensor with a depth of 1m. By comparing the difference between the events to the recordings, we can assume the time and provide explanations for the events. For instance, the decline in measurements on November 22nd, 2020, and April 11th, 2021, as well as May 28th, 2021, and March 23rd, 2022, was due to winter maintenance and broken sensors, respectively. Even minor changes, such as the sensor calibration on September 26th, 2020, can impact the data's usability and necessitate data cleaning procedures.

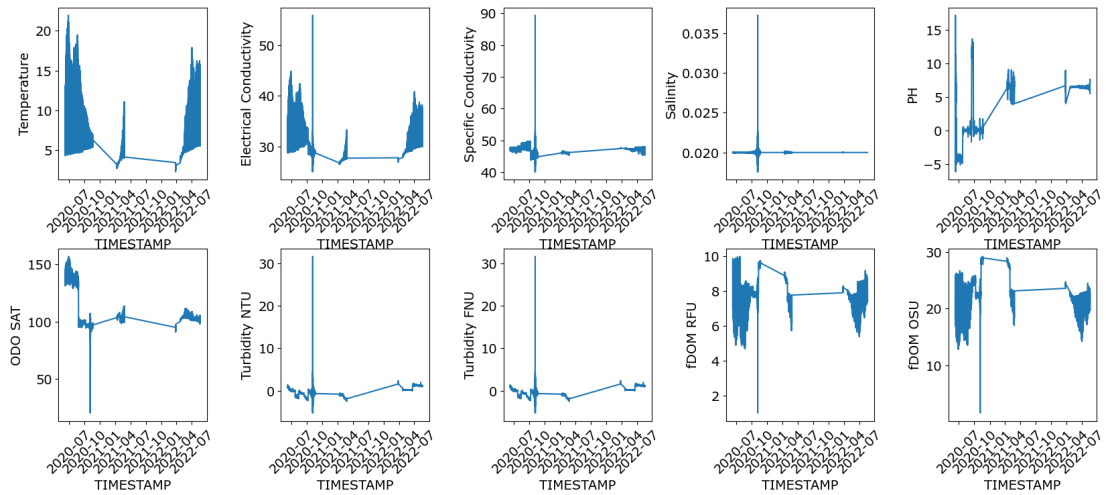


Figure 4.2: Time series visualization of sensor data

Table 4.1: Description of the raw dataset

Parameter	Count	Min	Max	Mean	Miss data
Timestamp	38675	20-06-09 15:14:06	22-08-19 01:18:29	/	0
Depth	38765	1	81	32.86	0
Temperature	38630	2.82	36.56	6.19	135
Salinity	38765	-4.35	12.08	0.02	0
PH	38765	-4.35	141.87	2.60	0
ODO SAT	38765	-1.39	155.73	110.98	0
Turbidity NTU	38765	-2.24	36.92	-0.25	0
Turbidity FNU	38765	0.836	81.036	32.86	0
fDOM RFU	38765	-0.08	25.85	8.24	0
fDOM OSU	38736	-1.69	29.22	23.41	29
Electrical Conductivity	38765	26.66	60.21	30.21	0
Specific Conductivity	38765	0.02	96.36	47.14	0

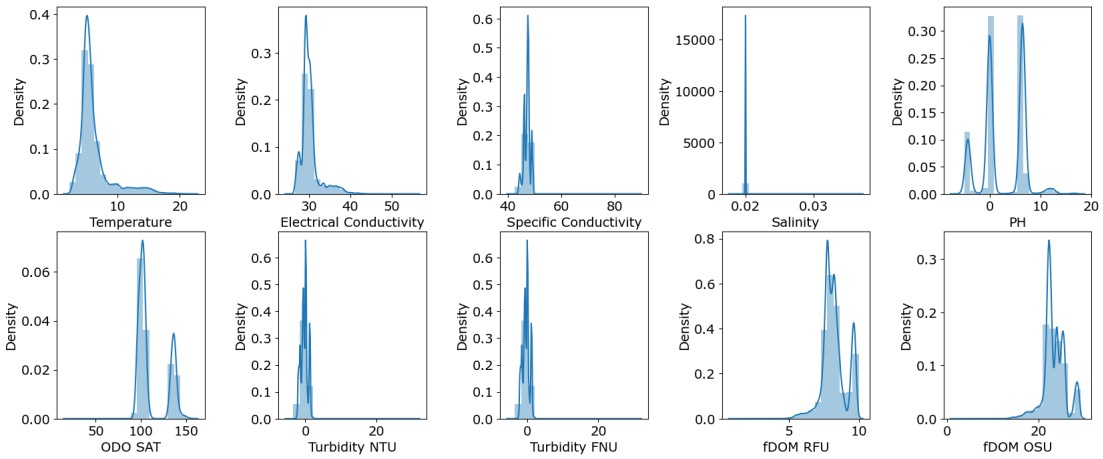


Figure 4.3: Distribution of flattened sensor data

Further analysis is conducted through the data distribution, as depicted in Figure 4.3. It can be observed that the temperature is distributed nearly normally, and the salinity is consistently at 0. Both distributions align with expectations. However, the other sensor data do not follow normal distributions, and PH has three peaks. According to the experts, this deviation is not always incorrect. With their assistance, the PH sensor and turbidity sensors are identified as the target sensors that require calibration, and the sensor error label is added based on their analysis as well. The objective of calibration is to make the sensor data as close to the clean data as possible in terms of distribution, mean value, and standard deviation.

The detection and correction of missing data are important in any data analysis project. However, in this project, only spiky data and drift data were considered for calibration. This is because the amount of missing data in the test dataset is small and the low sampling frequency of the sensor data, which is collected every 12 hours, makes missing data less of an issue. Therefore, the focus was primarily on correcting spiky and drift data, which have a greater impact on

the accuracy and reliability of the data analysis. Four labels were attached to the test data set where Group 0 represents no error, Group 1 represents spiky data, Group 2 represents drifted data, and Group 3 represents both types of errors.

4.2 Anomaly detection with CNN network

The labeled data were transformed into recurrence maps, as depicted in Figure 3.1 which are chosen randomly from the map pool. Two different mapping strategies were employed in this study. The first strategy considered the temporal dynamics of each sensor by retaining the original data values and calculating the recurrence matrix, as described in Section 3.2.1. The corresponding recurrence maps are presented in Figure 4.4.1. The second strategy involved the normalization of data from multiple sensors recorded at the same time. To mitigate the influence of varying scales across different sensor data, the data were normalized using Equation 4.1,

$$x' = \frac{x - X.mean}{X.std} \tag{4.1}$$

where X represents the data series, and x and x' denote the original and normalized data. Here, $X.mean$ and $X.std$ represent the mean value and standard deviation of the series, respectively. Through this normalization process, the shape of the distribution remained unchanged while the mean value moved to 0 and the standard deviation became 1. The recurrence maps for the normalized data are shown in Figure 4.4 .2, where a-d represents Groups 0-3, respectively. Notably, the color bar denotes different ranges in these maps, especially for spike data, where the range is larger than in other maps (Figure 4.4 .1.c and Figure 4.4 .2.c). A similar phenomenon was observed for the recurrence map with both errors, where the color bar range was the same, but the overall distance was larger than that of the data without errors. In other words, for spiky data, the difference between the two samplings was larger compared to other groups, which is consistent with the definition of spiky data.

After the maps are generated, they are fed into the CNN classifier for testing. Not only the loss line is depicted but also the accuracy is presented to evaluate the model's performance. While loss refers to the mathematical difference between the predicted output and true classification, accuracy focuses on how well the model can predict the correct label. Mathematically, it is the portion of correctly classified instances to the whole dataset. The results can be seen in Figure 4.4. Although for both strategies, the overall accuracy increases with the training epoch and finally reaches a relatively stable stage, the accuracy from the normalized multisensor is higher the unnormalized sole sensor. The former can reach as high as 99.6% while the latter can only reach 94.3%

To provide a more detailed comparison, a confusion matrix was utilized.Table 4.2 shows the detection class for every class and thus the confusion matrix which is presented in Table 4.3 can be generated. The F1 score, the harmonic mean of precision and recall, was applied to analyze this matrix, which can be expressed as follows:

$$F1(class = a) = \frac{1}{\frac{1}{Precision(class=a)} + \frac{1}{Recall(class=a)}} \tag{4.2}$$

where Precision(class=a) is defined as the ratio of true positive (TP) results to the sum of true

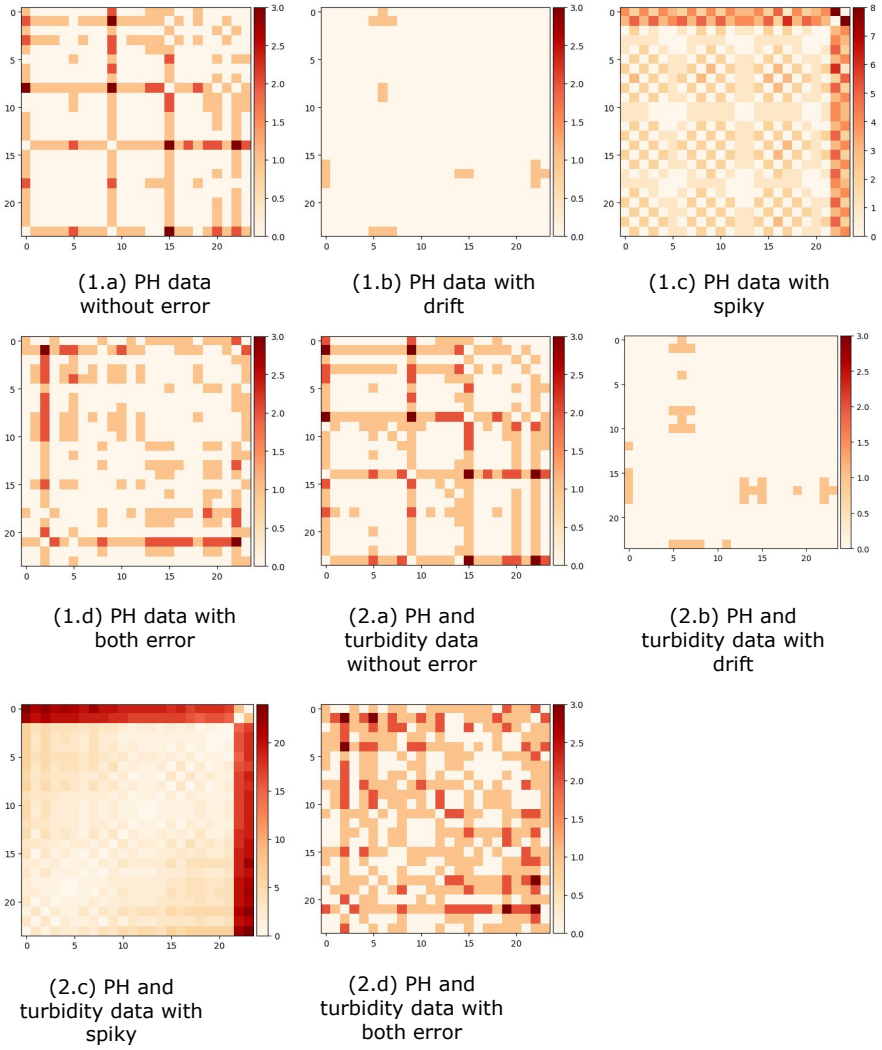


Figure 4.4: Time series visualization of sensor data

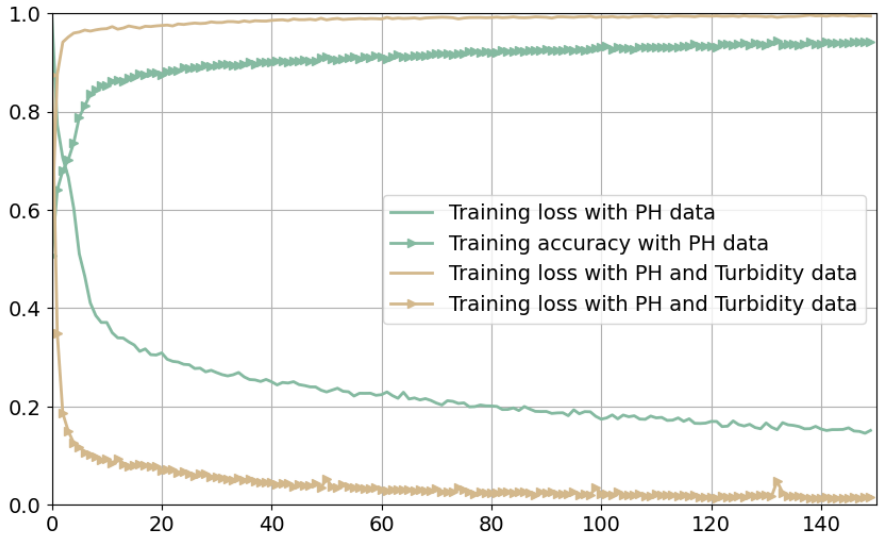


Figure 4.5: The training loss and accuracy change with epoch

Table 4.2: Anomaly detection results

	Class 1	Class 2	Class 3	Class 4
Class 1	2370	85	68	0
Class 2	7	3381	0	99
Class 3	56	0	2290	0
Class 4	0	120	0	2859

positive and false positive (FP) results for class a which measures the model's ability to identify positive results, and $Recall(class = a)$ is defined as the ratio of TP results to the sum of TP and false negative (FN) results for class a which measures the ability to capture all positive examples. It should be noted that these are based on a single class in a multiclass classification problem. In this case, the F1 score was calculated for each class separately as 0.96, 0.71, 0.97, and 0.96 for "clean", "drift", "spiky", and "both" class respectively.

Table 4.3: Confusion matrix for anomaly classification

Total samples= 11335		Prediction (%)			
		Clean	Drift	Spiky	Both
Ground truth	Clean	93.94	3.37	2.70	0
	Drift	0.20	96.96	0	2.84
	Spiky	2.39	0	97.61	0
	Both	0	4.03	0	95.97

4.3 Signal reconstruction comparison

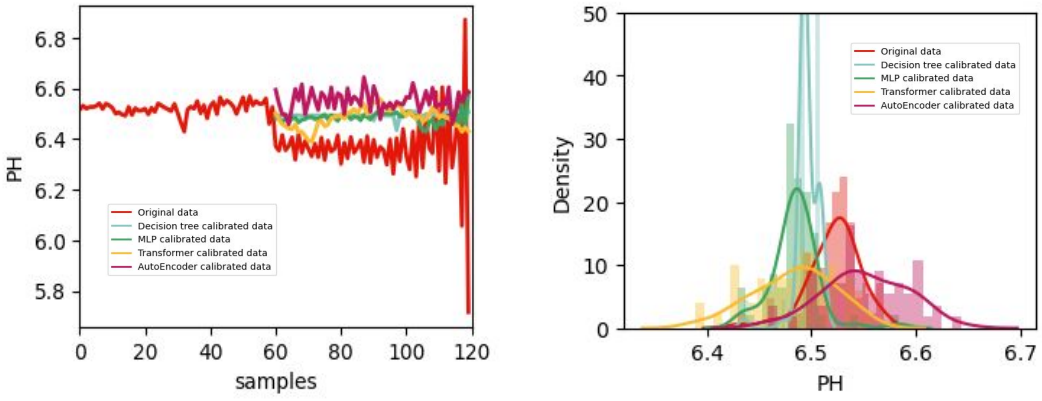
After detecting the underlying problem, four calibration methods, including the transformer network, MLP network, decision tree algorithm, and proposed convolutional AutoEncoder, were applied to generate high-quality data. According to the consultant with experts in this domain, three different sensors face problems in the collected dataset including one PH sensor and two Turbidity sensors which are chosen as the interested indicators. It should be noted that the water quality remained relatively consistent during the selected period, as per the current water quality monitoring program, and, therefore, the values from these sensors were expected to remain consistent as well. Based on this plus the lack of ground truth, the calibration methods' effectiveness was evaluated through data-driven analysis.

A more detailed analysis will be visualized based on a selected subset. In this work, a depth of 12 meters below the surface was chosen randomly for better visualization and analysis. It is worth noting that the same conclusion can be extended to other depths, given that the sensor platform, location, and data structure remain the same.

4.3.1 PH data calibration result comparison

The comparison of the PH data calibration results is depicted in Figure 4.6, with Figure 4.6(a) illustrating the regenerated time series data and Figure 4.6(b) presenting the corresponding data distribution.

Upon examining Figure 4.6(a), it is evident that the original data exhibit both drift and spikiness. After the 60th sample, the average value of the data decreases, while the variation starts to



(a) Reconstructed time series signal data for PH

(b) Density distribution of reconstructed PH data

Figure 4.6: Comparison of calibration results for turbidity PH data

increase. The figure also showcases the time dynamics obtained from the calibration algorithms. The transformer and AutoEncoder algorithms produce a broader range of regenerated values compared to the other algorithms. Notably, the decision tree algorithm yields only four distinct values, resulting in a limited prediction range that is less consistent with reality. This observation is further supported by the density distribution depicted in Figure 4.6(b), where the decision tree method exhibits a higher peak compared to all the other algorithms.

Conversely, the MLP, Transformer, and AutoEncoder models provide more realistic results. In terms of spikiness, all the calibration methods generate data with reduced spikes. However, when considering drift, although some degree of drift remains compared to the original clean data, it is significantly reduced. Among these three approaches, the AutoEncoder produces a dataset that closely resembles the clean data.

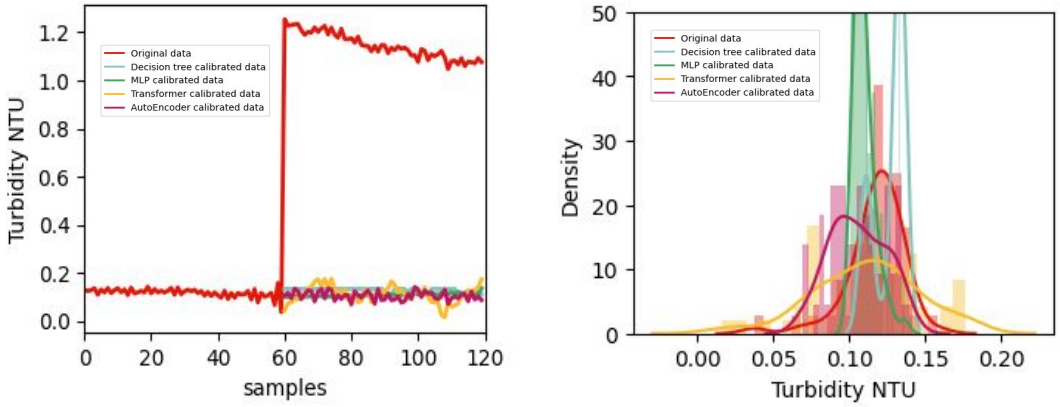
Consequently, all three methods yield higher-quality data, with the proposed AutoEncoder approach demonstrating the best results from a data analysis perspective.

4.3.2 Turbidity NTU data calibration result comparison

Figure 4.7 provides a visual representation of the calibration results for the turbidity measured in Nephelometric Turbidity Units (NTU). One of the primary challenges encountered during the calibration process is the presence of drift, which manifests as a significant increase in all values after the 60th sample. Consequently, an effective calibration algorithm must address and minimize this drift phenomenon.

The calibration algorithms employed in this study have demonstrated varying degrees of success in reducing drift, as depicted in Figure 4.7(a). Notably, the decision tree algorithm yields results similar to the PH calibration, wherein only a limited range of possible values is observed. Similarly, the MLP model generates regenerated data within a narrow data range, resulting in minimal changes in the time series plot (as seen in Figure 4.7(a)) and a prominent peak in the data distribution plot presented in Figure 4.7(b).

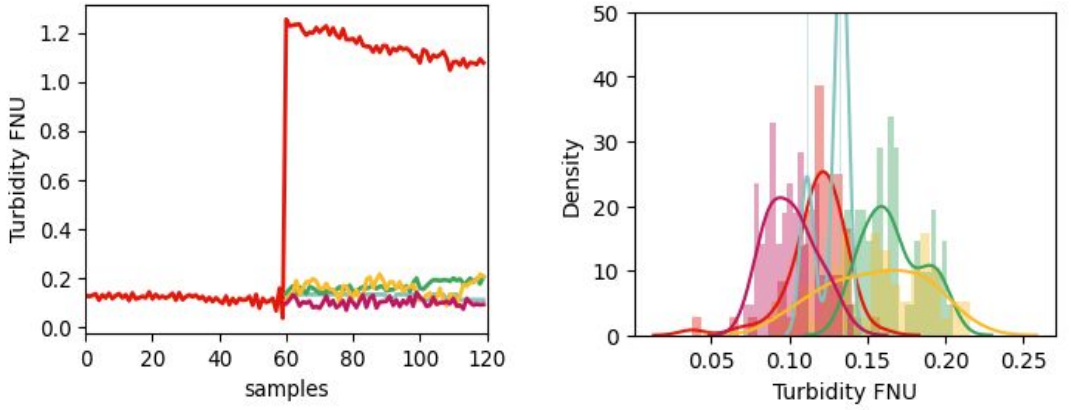
On the other hand, the transformer network, which primarily analyzes the temporal dynamics



(a) Reconstructed time series signal data turbidity NTU

(b) Density distribution of reconstructed turbidity NTU data

Figure 4.7: Comparison of calibration results for turbidity Turbidity NTU data



(a) Reconstructed time series signal data for Turbidity FNU

(b) Density distribution of reconstructed Turbidity FNU data

Figure 4.8: Comparison of calibration results for turbidity FNU data

underlying the sensor data, and the AutoEncoder, which incorporates both temporal dynamics and sensor correlation, produce data with closer distribution patterns and temporal dynamics. While the mean value and standard deviation may not precisely match the values of the clean data, they exhibit greater proximity compared to the results generated by the other two algorithms.

Based on these findings, both the transformer network and the AutoEncoder can be considered successful approaches for calibrating turbidity NTU.

4.3.3 Turbidity FNU data calibration result comparison

The final problematic sensor measures the turbidity in Formazin Nephelometric Units (FNU), which is correlated to the turbidity measured in Nephelometric Turbidity Units (NTU). Thus, the problem and target remain unchanged in this context. The comparison of the calibration results for turbidity FNU data is depicted in Figure 4.8.

In theory, the results for both sensors should be similar. However, there are several noteworthy observations even after applying various calibration algorithms to eliminate or reduce drift. Firstly, while the decision tree algorithm still yields comparable results due to the limited range of possible data, the MLP model demonstrates improved performance compared to the calibration for turbidity NTU. With both a closer mean value and standard deviation to the clean data, the MLP model proves to be a more favorable solution than the transformer model. Secondly, it is worth mentioning that although the transformer model can accomplish the task to some extent, errors still accumulate, resulting in larger deviations. Among all the results, the AutoEncoder method generates the most optimal dataset. Although the drift is still present, it is considerably smaller than in the other three calibrated datasets, and the distribution closely resembles a normal distribution, similar to the clean data.

4.3.4 Calibration result comparison overview

Based on the analysis conducted in the previous subsections, the performance evaluation of different methods reveals interesting findings. The AutoEncoder approach consistently demonstrates superior performance across all three indicators, while the transformer network shows comparable results specifically for the turbidity NTU parameter. However, it is important to consider the mathematical distribution of the results, as this provides additional insights.

To facilitate comparison, Table 4.4 presents a comprehensive analysis of the distribution of clean data and calibrated data obtained from three different methods: Decision tree, MLP, and Transformer. The table showcases the mean value and standard deviation for each sensor, serving as key criteria for evaluation.

Table 4.4: Distribution comparison among clean data and calibrated data from four methods

Source \ Criteria	PH mean	PH std	T-NTU mean	T-NTU std	T-FNU mean	T-FNU std
Clean	6.360	0.034	0.119	0.072	0.126	0.012
Decision tree	6.425	0.044	0.124	0.005	0.124	0.005
MLP	6.502	0.043	0.125	0.012	0.138	0.018
Transformer	6.383	0.039	0.111	0.035	0.155	0.032
AutoEncoder	6.350	0.040	0.102	0.017	0.101	0.017

It is worth noting that the purpose of calibration algorithms extends beyond the reconstruction of time series data. An ideal algorithm should not only generate data with similar mean values but also exhibit a standard deviation that brings the distribution closer to that of clean data. Analyzing the results, it becomes evident that all methods display a larger standard deviation for pH compared to the original clean data. However, the transformer network achieves the closest standard deviation, while the AutoEncoder model regenerates the mean value that closely resembles the original data. It is important to consider the disparity between the mean value offset (only 43.5% for AutoEncoder compared to transformer) and the standard deviation offset (which amounts to 83.3% for the transformer model). Thus, based on this evaluation criterion, the AutoEncoder approach demonstrates the most favorable performance.

On the other hand, different conclusions arise when assessing the turbidity NTU parameter. The MLP network exhibits a mean value and standard deviation that are closer to the clean data compared to other methods. Conversely, the AutoEncoder produces results that deviate

significantly from the clean data, even when compared to the decision tree and transformer. Therefore, the MLP approach is considered the most suitable for calibration in this regard.

Regarding turbidity FNU, the transformer model performs less effectively than the other three models, as evidenced by a considerably larger offset in the mean value. While the decision tree algorithm generates a mean value that closely resembles the clean data, the AutoEncoder exhibits a standard deviation that is closer to the desired outcome. Overall, the MLP method produces the most satisfactory results, with a mean value offset of only 9.5% and a standard deviation offset of 50.0%.

Although these analyses indicate that the PH sensor should be calibrated using the AutoEncoder method and the turbidity sensors are best calibrated with MLP, it is important to acknowledge that further analysis is required with the clustering algorithm. The clustering algorithm will generate the final results for the project, which will subsequently be evaluated based on the quality of the clustering outcomes. Therefore, additional investigations are necessary to fully assess the effectiveness of the chosen calibration methods.

4.4 Clustering

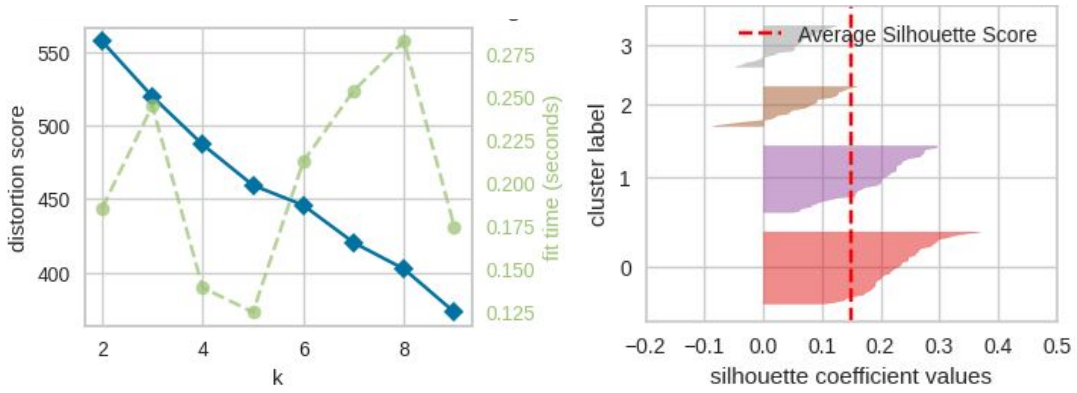
With these calibrated data, clustering algorithms can be applied to generate the final clustering for water quality monitoring. Besides the originally clean data, the calibrated data are also taken into consideration. The combination will then be fed into the clustering algorithm to evaluate the final results for our project.

4.4.1 K-means clustering result analysis

To determine the optimal number of clusters for the K-means algorithm, an elbow graph is employed, which assesses the relationship between clustering quality and the number of clusters. The x-axis of the graph represents the number of clusters, while the y-axis denotes evaluation metrics, specifically the distortion score and fit time in this particular project. It is crucial to consider the project's characteristics, such as the dataset size and runtime. Currently, the project comprises only 126 samples, resulting in a relatively short runtime. However, as the dataset accumulates more samples, runtime becomes an increasingly important factor that necessitates careful consideration.

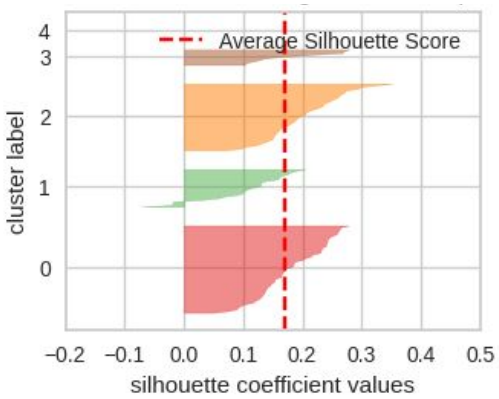
The primary objective of the elbow graph is to identify the point at which further increases in the number of clusters fail to yield significant improvements in clustering quality. This point is commonly referred to as the "elbow," representing the juncture where the curve starts to flatten out. However, it is crucial to acknowledge that the location of the elbow is reliant on the specific objectives and characteristics of the dataset. It is essential to supplement the analysis with domain knowledge, as the elbow graph alone provides a solution solely from a data-centric perspective and may not be directly applicable in real-life scenarios.

In previous work, we applied 4 different kinds of algorithms to conduct the calibration task and their results are compared in this part. Figure 4.9, Figure 4.10, Figure 4.11, and Figure 4.12 represents the clustering analysis results with transformer, MLP, decision tree, and AutoEncoder calibrateed data respectively.



(a) Evaluation for clustering with transformer calibrated data

(b) CLuster evaluation when K = 4



(b) CLuster evaluation when K = 5

Figure 4.9: Clustering results with K-means using transformer calibrated data

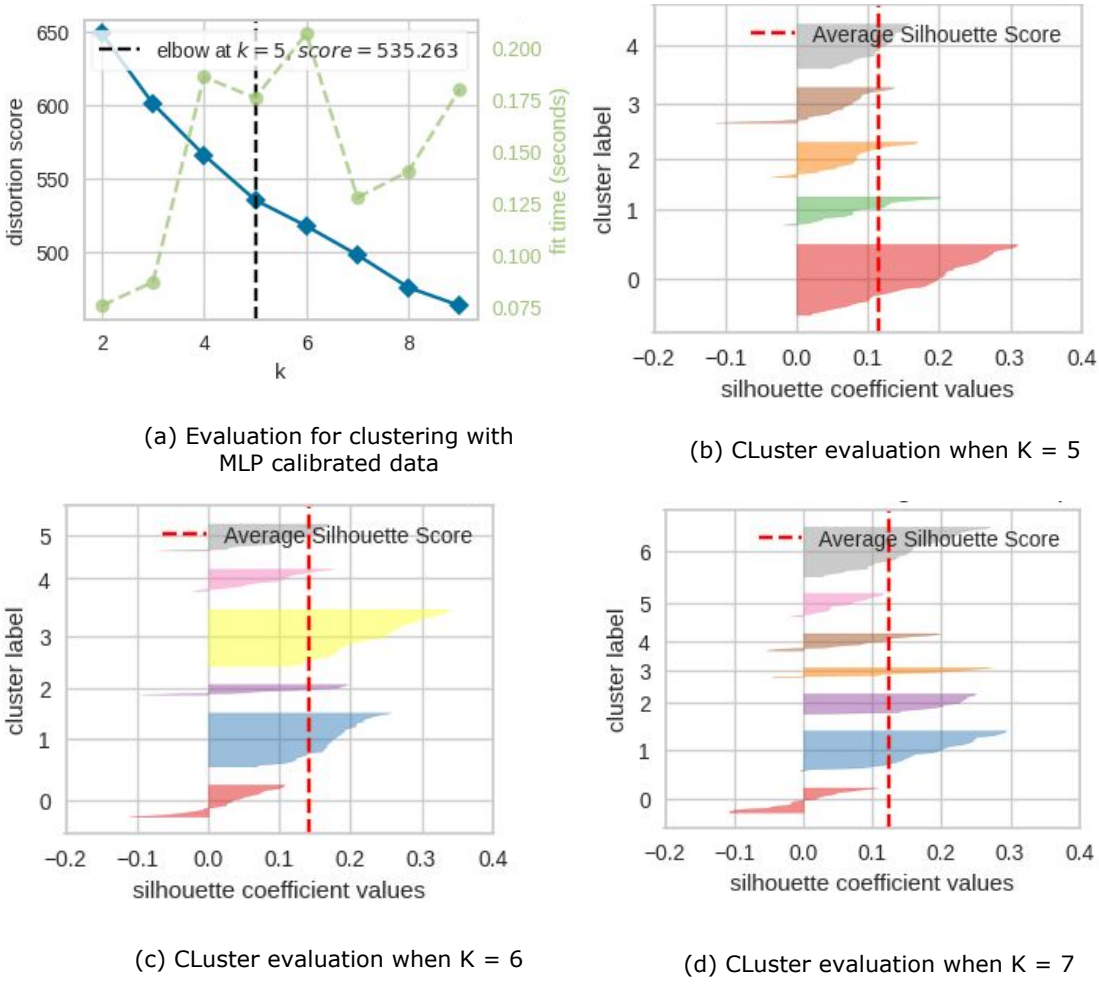
Figure 4.11(a) illustrates the elbow graph for transformer-calibrated data, revealing the absence of a distinct elbow point. However, it is worth noting that the algorithm's performance exhibits a significant runtime reduction at 4 and 5 clusters without a substantial drop in quality. Consequently, these two cluster numbers are considered as potential candidates for providing optimal solutions. To delve deeper into the analysis, Silhouette plots are employed.

The Silhouette plot, represented in Figure 4.11(b) and Figure 4.11(c), portrays the shape of the data and assigns different colors to distinguish between clusters along the y-axis. The red dotted line corresponds to the average score of the entire dataset. According to [68], besides having higher Silhouette scores, an ideal choice for K would result in all clusters having Silhouette scores higher than this average line, while simultaneously minimizing fluctuations in the data size. Upon examining the Silhouette plots, it becomes evident that neither the 4-cluster configuration nor the 5-cluster configuration represents the optimal solution. This conclusion is based on the observation that Cluster 3 when K is set to 4 displays lower Silhouette scores compared to the average score and Cluster 2 and Cluster 0 have more data points than the other groups when K is set to 5.

The same analysis was performed on the MLP calibrated data, as depicted in Figure 4.10. In contrast to the previous method, this dataset exhibits a more evident elbow point when K is set to 5. To visualize the K-means clustering results, Figure 4.10(b) is presented. However, despite this elbow point, the clustering outcome is not optimal, primarily due to the imbalance in Cluster 0, which contains a significant number of data points that could potentially be subdivided into smaller groups. As a result, further experimentation was conducted by testing K values of 6 and 7. Regrettably, these additional cluster numbers did not lead to substantial improvements in the clustering results. When K is set to 6, Cluster 1 and Cluster 3 dominate the distribution, while Cluster 0 continues to exhibit a lower score than the average. Consequently, a 6-cluster configuration does not yield a satisfactory solution. Considering the distribution is more evenly balanced in the 5-cluster configuration, it is deemed a preferable option over the 6-cluster solution.

The results obtained from the decision tree calibrated data are presented in Figure 4.11(a). Similar to the transformer dataset, no distinct elbow point is observed in this case. However, it is noteworthy that the runtime exhibits local minima at K values of 3, 6, and 8. Consequently, these specific cluster numbers are further investigated by comparing the Silhouette plots to determine the optimal choice. Upon analyzing them, it becomes evident that a K value of 3 yields the least favorable results. Not only does it have one cluster that fails to reach the average score, but Cluster 1 also exhibits a larger dataset size compared to the other two clusters. Conversely, when considering the 6-cluster configuration, all the clusters reached the average score suggesting a better clustering outcome. However, it is worth noting that the 8-cluster configuration displays an even smaller variation in the distribution of data size, making it a preferable choice. Taking all these factors into account, the 8-cluster configuration can be considered as the optimal choice for the decision tree calibrated data.

Lastly, we analyze the results obtained from the AutoEncoder calibrated data, as illustrated in Figure 4.12. Examining subfigure (a), we observe an apparent elbow point at 3 clusters. Notably, this configuration achieves a silhouette score as high as 0.5, surpassing all previous clustering attempts. However, it is important to consider the individual cluster scores in relation to the average score. Among the clusters, only Cluster 1, encompassing over half of the data, exhibits a higher score than the average. Upon further evaluation, it is determined that a 5-cluster



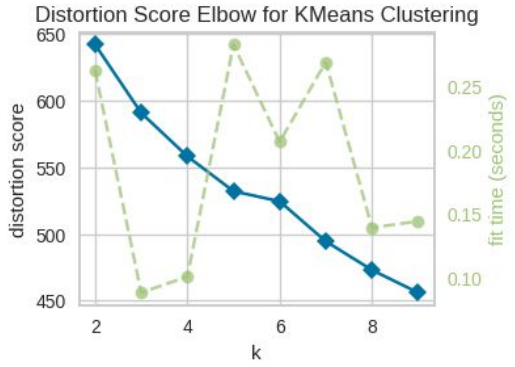
(a) Evaluation for clustering with MLP calibrated data

(b) CLuster evaluation when K = 5

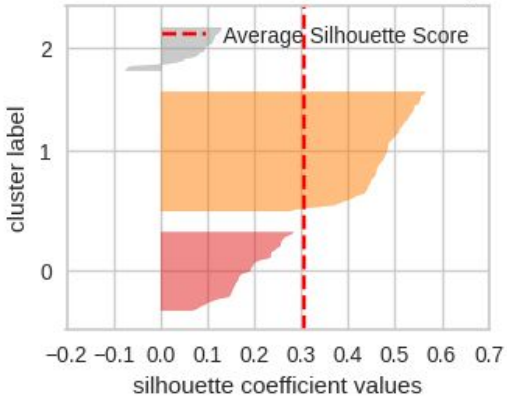
(c) CLuster evaluation when K = 6

(d) CLuster evaluation when K = 7

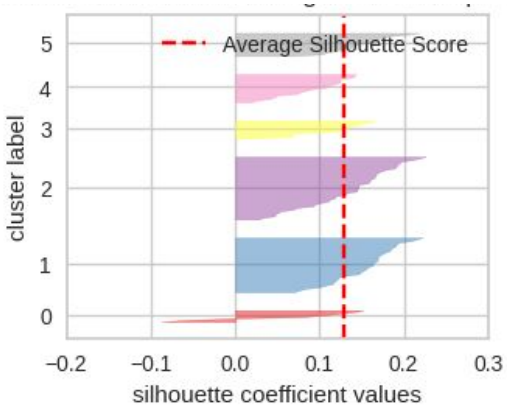
Figure 4.10: Clustering results with K-means using MLP calibrated data



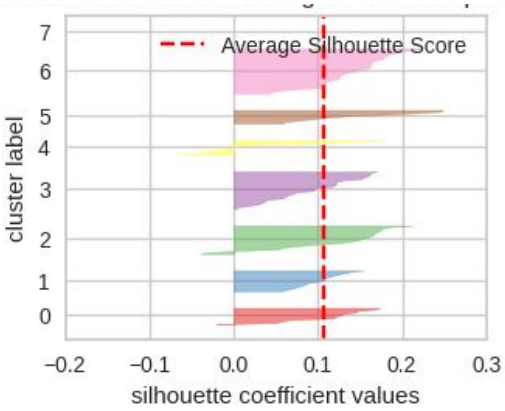
(a) Evaluation for clustering with decision tree calibrated data



(b) CLuster distribution when K = 3

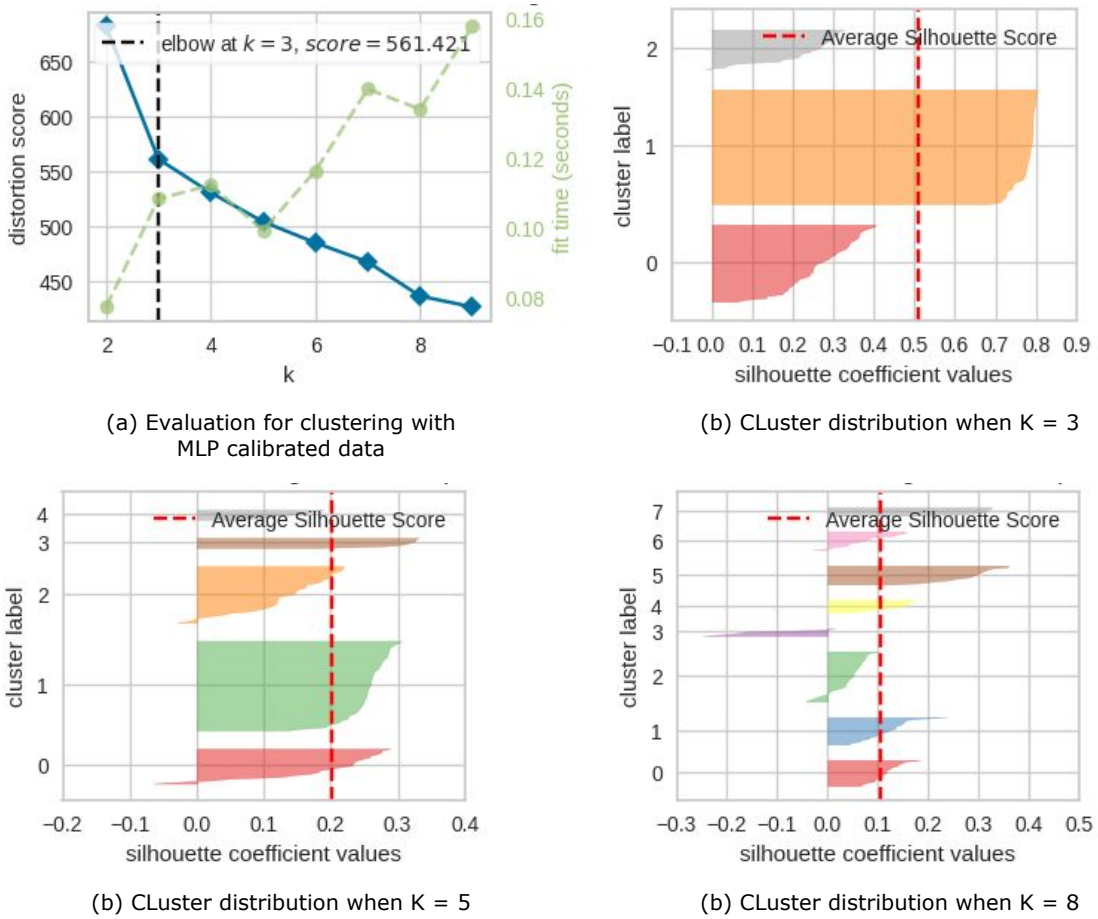


(b) CLuster distribution when K = 4



(b) CLuster distribution when K = 5

Figure 4.11: Clustering results with K-means using decision tree calibrated data



(a) Evaluation for clustering with MLP calibrated data

(b) CLuster distribution when K = 3

(b) CLuster distribution when K = 5

(b) CLuster distribution when K = 8

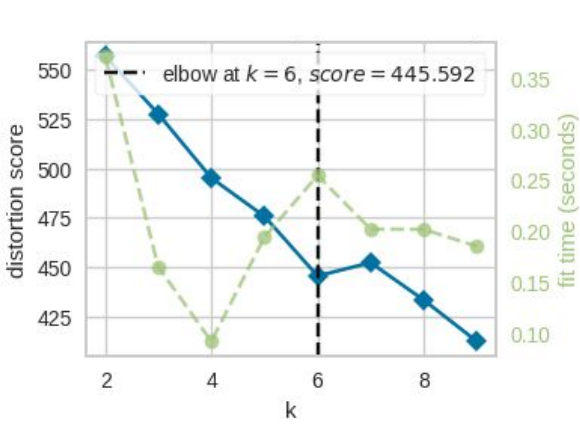
Figure 4.12: Clustering results with K-means using AutoEncoder calibrated data

configuration is more favorable than 8 clusters. This conclusion arises from the observation that Cluster 3 fails to reach the average score, and Cluster 2 barely achieves baseline performance. However, even though Cluster 5 is not ideal either, it is imperative to consider the distribution of data among clusters. Cluster 1 retains a larger data size than the other groups, while Cluster 4 contains too few data points. To explore potential alternative cluster numbers, configurations ranging from 2 to 9 clusters were tested. However, none of these configurations outperformed the 5-cluster arrangement in terms of both score and data distribution. Consequently, we select 5 as the optimal number of clusters for the AutoEncoder calibrated dataset, considering its satisfactory score and relatively balanced data distribution.

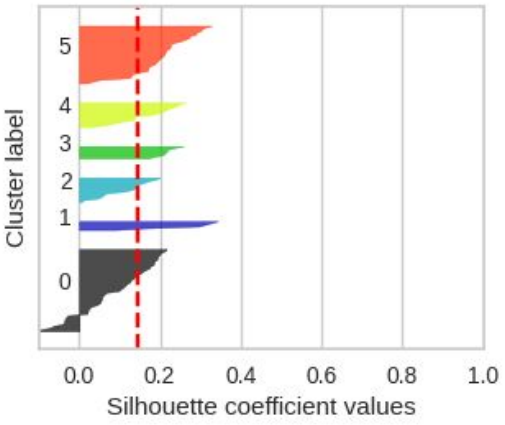
4.4.2 Spectral clustering result analysis

In the evaluation of spectral clustering, the elbow graph and the silhouette score are adopted. The elbow graph is utilized as a preliminary step to identify potential cluster numbers, while the silhouette plot serves as a more detailed assessment. Combining these two evaluation techniques, the spectral clustering algorithm can be effectively assessed in terms of determining the optimal number of clusters and evaluating the overall clustering performance.

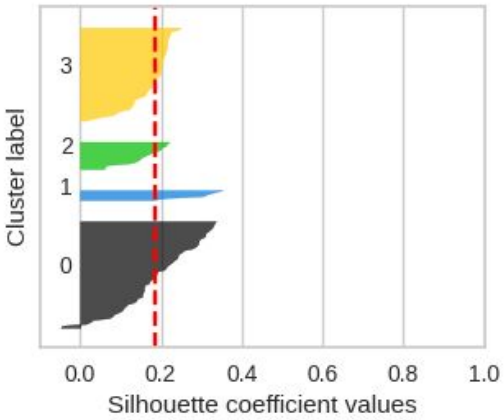
The analysis of the transformer-calibrated dataset using spectral clustering is presented in Fig-



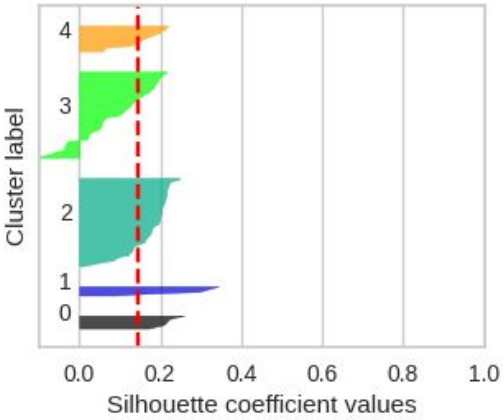
(a) Evaluation for spectral clustering with decision tree calibrated data



(b) Silhouette plot when K = 6



(b) Silhouette plot when K = 4



(b) Silhouette plot when K = 5

Figure 4.13: Clustering results with spectral clustering using transformer calibrated data

ure 4.13. Figure 4.13(a) shows that the most promising clustering outcome is achieved when the cluster number is set to 4. The corresponding silhouette plot is depicted in Figure 4.13(b). It is observed that Cluster 0 and Cluster 5 contain a larger number of data points, while the remaining four clusters exhibit a similar distribution of data points. To thoroughly evaluate alternative cluster numbers, we examined values ranging from 2 to 9. However, only cluster numbers 4 and 5 yielded results in which all clusters attained a higher silhouette score than the average. These results are shown in Figure 4.13(b) and Figure 4.13(c). Nonetheless, both cluster numbers 4 and 5 were deemed inferior to the cluster number 6. This decision was based on the observation that the dominant cluster became more apparent in the former cases, while the cluster number 6 exhibited a more balanced distribution of data points across clusters.

The analysis of the MLP calibrated data is presented in Figure 4.14. Unlike the previous cases, no clear elbow point is observed in the elbow graph, indicating that the determination of the number of clusters relies heavily on the silhouette plots. Among the tested cluster numbers (2, 4, 5, and 9), those that had at least one cluster with a silhouette score lower than the average score of the entire dataset were excluded from further consideration. The results for the remaining four cluster numbers are compared and evaluated. When the cluster number is set to 3, Cluster 2 exhibits a significantly lower number of data points compared to the other two clusters, making

it an unfavorable choice. On the other hand, the data distribution appears more even for the other three cluster numbers. Although two clusters consistently have a larger data size than the remaining clusters (e.g., Cluster 0 and 3 for K = 6, Cluster 0 and 1 for K = 7, and Cluster 1 and 5 for K = 8), the distribution of data points among the other clusters does not deviate significantly. Additionally, the average silhouette scores for all three cluster numbers are identical (0.11), providing no clear guidance for determining the best number of clusters based on this criterion alone. However, upon revisiting the distortion and runtime metrics, it is observed that the cluster number 8 outperforms the other two options, exhibiting lower runtime and distortion. Taking these factors into consideration, cluster number 8 is selected as the optimal choice.

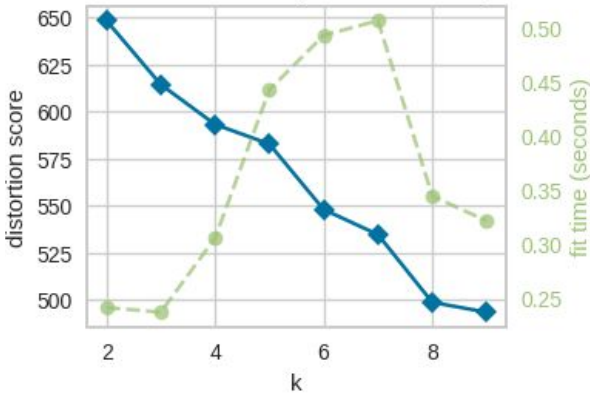
The findings obtained from the analysis of decision tree calibrated data are presented in Figure 4.15.

In Figure 4.15(a), the elbow graph reveals an elbow point at 4 clusters. However, a closer examination of the silhouette plot depicted in Figure 4.15(b) indicates an unequal distribution of data points among the clusters. This prompts us to explore whether a more optimal cluster number can be identified. To investigate this, we conducted a thorough evaluation by considering cluster numbers ranging from 2 to 9. Specifically, we focused on clusters that yielded silhouette scores for every cluster higher than the average. The results of these evaluations are illustrated in Figure 4.15(c) to (f). It is evident that increasing the number of clusters does not effectively address the variation in data distribution. Notably, when the cluster number is set to 5, the variation in data distribution even becomes more pronounced. Furthermore, when comparing the cluster numbers of 6 and 7, it is observed that Cluster 5, which existed in the 6-cluster scenario, splits into two separate clusters in the 7-cluster scenario. However, this division does not significantly improve the clustering quality, while concurrently increasing the computational time. Moreover, the data distribution across clusters in the 6-cluster scenario appears to be more evenly distributed than in the 4-cluster scenario, suggesting that 6 clusters may offer a more suitable clustering solution.

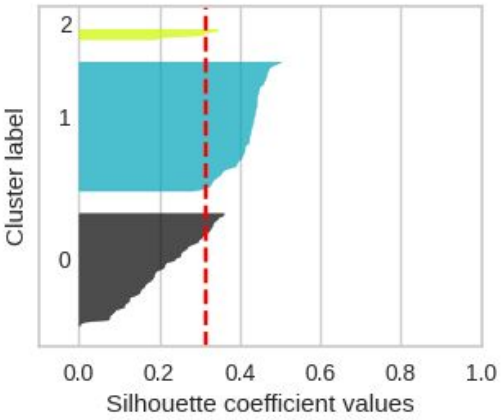
The analysis of the AutoEncoder calibrated data is depicted in Figure 4.16. The elbow graph indicates that the optimal number of clusters is 4 (K=4), as shown in the subplot (b). However, it is important to note that the resulting distribution of clusters is not perfect. In addition to considering the elbow point, other cluster numbers were evaluated based on their comparison to the average silhouette score. It was observed that none of the alternative cluster numbers provided satisfactory results when compared to the average silhouette score. Therefore, 4 clusters (K=4) were selected as the most suitable option based on this criterion.

4.4.3 SOM clustering result analysis

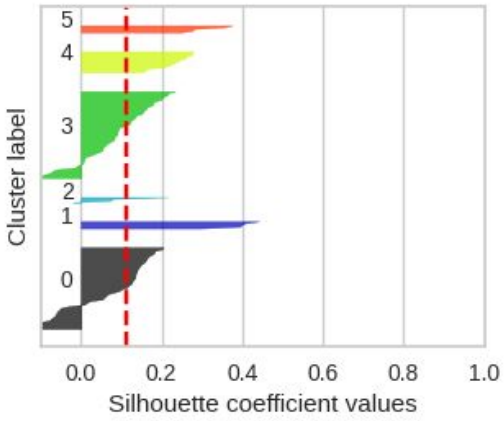
Figure 4.17 illustrates the outcomes obtained from applying the SOM clustering algorithm to four calibrated datasets. These heatmaps enable a comparison of the silhouette scores based on different SOM shapes, aiding in the determination of optimal parameters. It is noteworthy that the highest silhouette scores are consistently observed at coordinates $(2, 1)$, indicating a tendency for samples to be clustered into two groups. However, this preference for fewer clusters stems from the silhouette score's inclination towards selecting configurations with minimal dissimilarity within clusters and maximal dissimilarity between clusters, rather than solely considering the number of clusters. Consequently, it is crucial to consider alternative shapes that yield high scores.



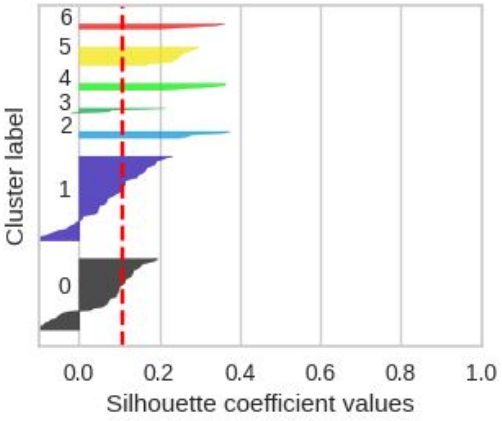
(a) Evaluation for spectral clustering with MLP calibrated data



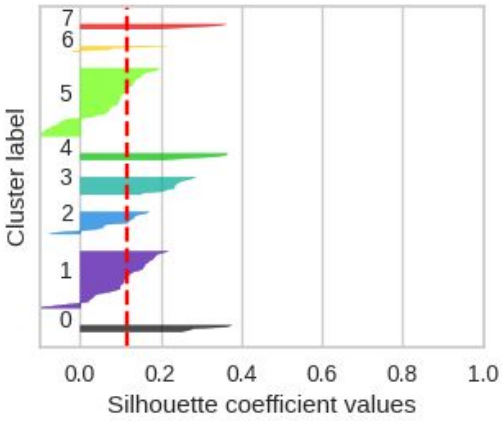
(b) Silhouette plot when K = 3



(c) Silhouette plot when K = 6

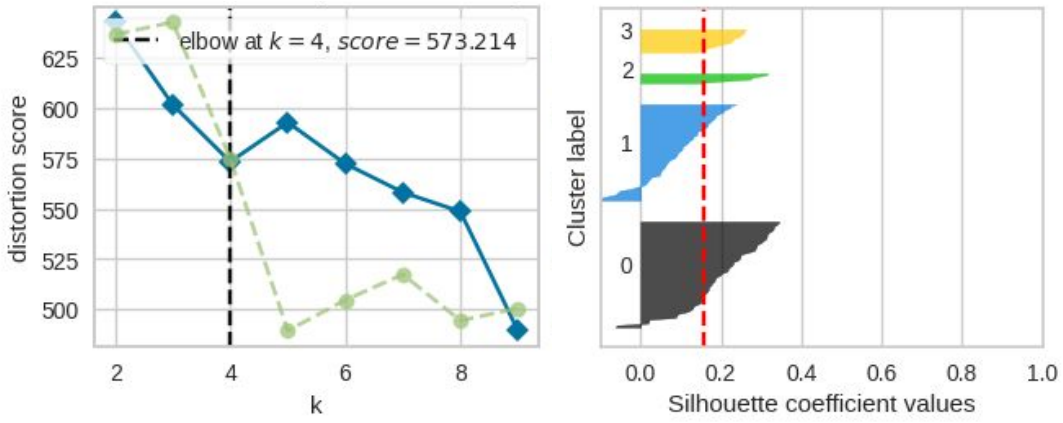


(d) Silhouette plot when K = 7



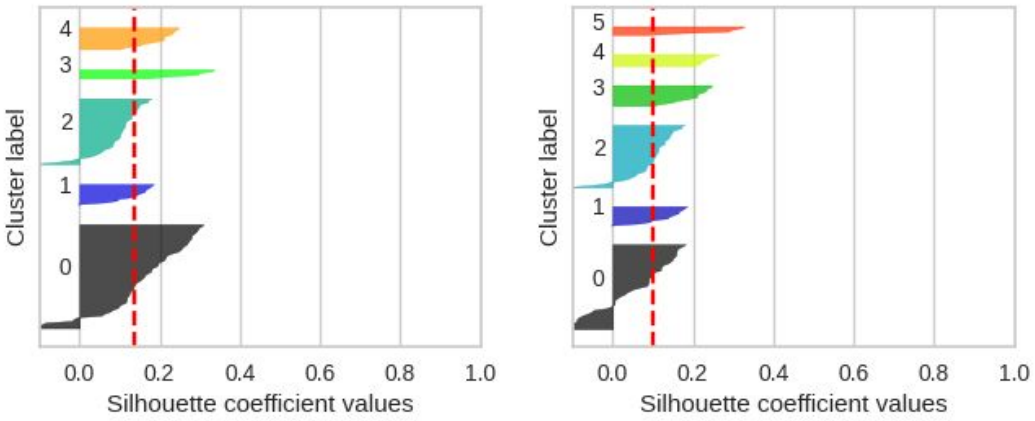
(e) Silhouette plot when K = 8

Figure 4.14: Clustering results with spectral clustering using MLP calibrated data



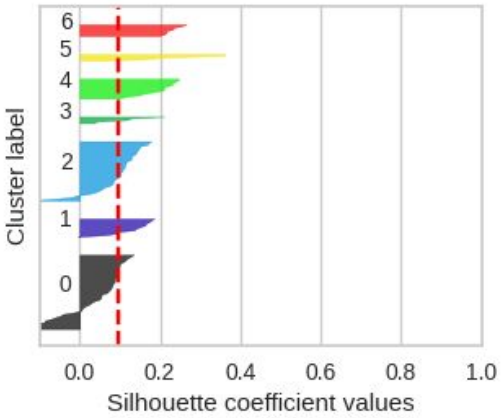
(a) Evaluation for spectral clustering with decision tree calibrated data

(b) Silhouette plot when K = 4



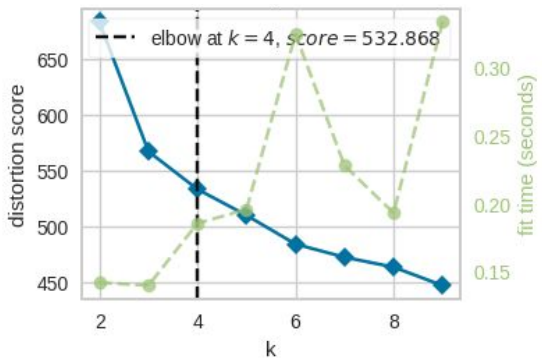
(c) Silhouette plot when K = 5

(d) Silhouette plot when K = 6

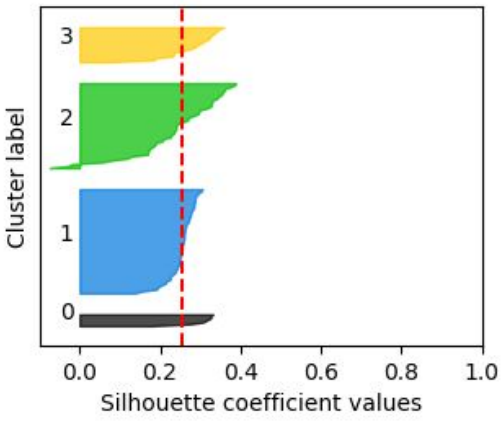


(e) Silhouette plot when K = 7

Figure 4.15: Clustering results with spectral clustering using decision tree calibrated data

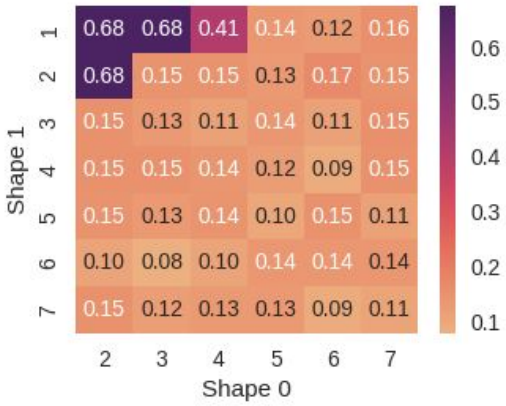


(a) Evaluation for spectral clustering with AutoEncoder calibrated data

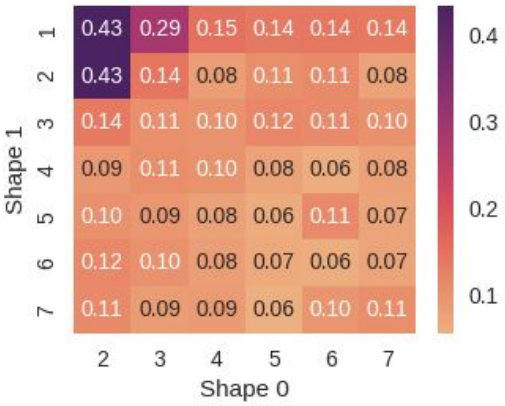


(b) Silhouette plot when K = 4

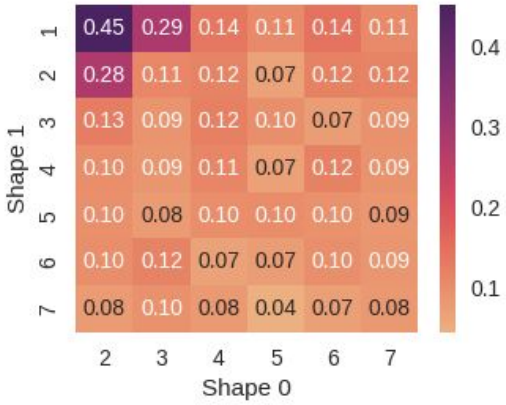
Figure 4.16: Clustering results with spectral clustering using decision tree calibrated data



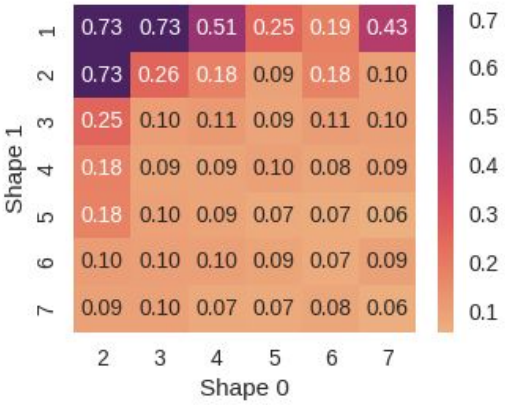
(a) Silhouette score heatmap for shape analysis with transformer calibrated data



(b) Silhouette score heatmap for shape analysis with MLP calibrated data



(c) Silhouette score heatmap for shape analysis with decision tree calibrated data



(d) Silhouette score heatmap for shape analysis with AutoEncoder calibrated data

Figure 4.17: Heatmap for the shape of SOM and silhouette score

Analyzing the transformer-calibrated data depicted in Figure 4.17(a), we observe identical scores at coordinates $(2, 1)$, $(3, 1)$, and $(2, 2)$. Notably, these shapes produce similar clustering results, partitioning the data into two groups. Furthermore, the shape $(4, 1)$, with a score of 0.41 (the second highest), also generates identical results. Moving to Figure 4.17(b), which represents the MLP-calibrated data, we find that only shapes $(2, 1)$ and $(2, 2)$ yield the highest scores and thus warrant further evaluation. While the former shape exhibits the same distribution as the transformer-calibrated data, the latter divides the data into three distinct groups.

Continuing the analysis with the decision tree calibrated data shown in Figure 4.17(c), it becomes evident that shape $(2, 1)$ achieves a higher score (0.45) compared to other configurations. However, shape $(3, 1)$, despite its lower score of 0.29, could also be considered as a potential optimal shape. Finally, Figure 4.17(d) presents the results obtained from the AutoEncoder calibrated data. Similar to the transformer results, shapes $(2, 1)$, $(3, 1)$, and $(2, 2)$ exhibit identical silhouette scores. Among these, shape $(2, 2)$ clusters the data into three distinct groups.

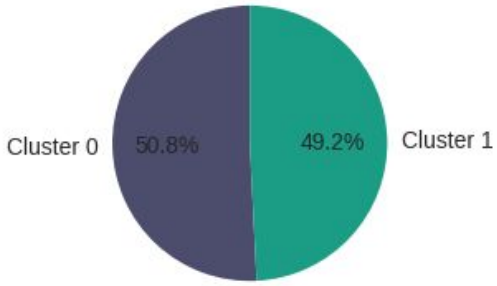
These findings highlight the importance of exploring various SOM shapes beyond the highest-scoring configurations. By considering alternative shapes, researchers can gain valuable insights into the clustering patterns and potentially uncover more nuanced structures within the data.

The clustering distribution is visualized in Figure 4.18. In subplot (a), the results from the transformer-calibrated data exhibit a two-group division, with each group containing approximately half of the data points. This two-cluster pattern is also observed in the other datasets with a shape of $(2, 1)$, hence they are not presented in this figure. Moving to subplot (b), the results obtained from the MLP-calibrated data showcase a $(2, 2)$ shape. Although three clusters are formed, the third cluster accounts for only 0.8% of the total data points, leading to a resemblance to the transformer results. The quality of clustering improves when using decision tree-calibrated data, as illustrated in Figure 4.18(c), and AutoEncoder-calibrated data, shown in Figure 4.18(d). In these cases, the third clusters consist of a greater proportion of data points. While both Cluster 0 groups have the same ratio of data points (49.2%), the AutoEncoder model assigns 23.0% of the data to cluster 3, resulting in a more balanced data distribution compared to the decision tree model.

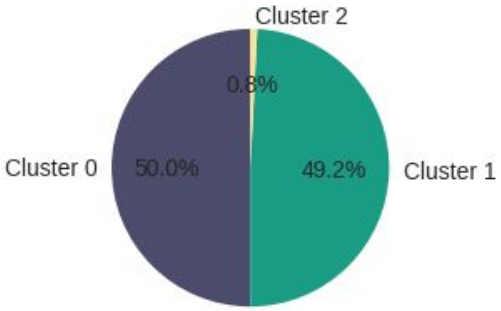
4.5 Water quality clustering comparison

The final results obtained from the clustering model using differently calibrated data are compared using Table 4.5, Table 4.7, and Table 4.6 in terms of the number of clusters, the standard deviation of cluster distribution and silhouette scores. Instead of evaluating the calibration model and clustering model separately, we analyzed the final results that take both aspects into account.

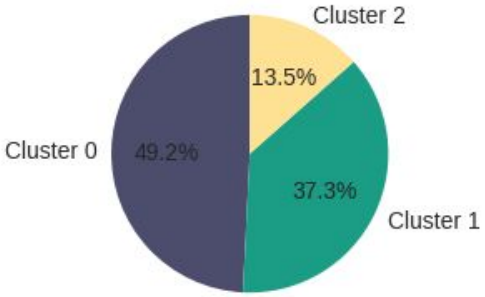
The optimal number of clusters is influenced by both the calibration and clustering algorithms as shown in Table 4.5. It is worth noting that although the number of clusters can range from as high as 8 (K-means with decision tree calibrated data and spectral clustering with MLP calibrated data) to as low as 2 (SOM with transformer calibrated data), different clustering algorithms have varying dominant numbers. For example, for K-means, the optimal value is 5 for three different datasets, while for SOM, 3 is the most favorable since all MLP-, decision tree-, and AutoEncoder-calibrated data yield an optimal number of 3. On the other hand, spectral clustering prefers 6. This difference not only affects the final clustering results but also impacts



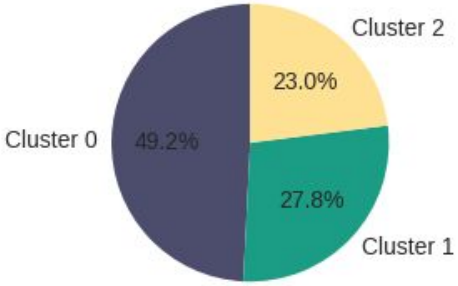
(a) Clustering distribution with SOM using transformer calibrated data



(b) Clustering distribution with SOM using MLP calibrated data



(c) Clustering distribution with SOM using decision tree calibrated data



(d) Clustering distribution with SOM using AutoEncoder calibrated data

Figure 4.18: Clustering distribution for the SOM clustering results using differently calibrated data

the performance of the models like the cluster distribution and silhouette score used in this project.

Table 4.5: The number of clusters for each combination from this project

algorithms	Transformer	MLP	Decision Tree	AutoEncoder
K-means	5	5	8	5
Spectral	6	8	6	4
SOM	2	3	3	3

From Table 4.6, it is observed that an increase in the number of clusters does not necessarily result in a decrease in distortion, which serves as an indicator of improved clustering results. This is because the improved quality brought by increasing cluster numbers is only achieved when a larger data cluster is divided into smaller ones. However, in some cases, the opposite occurs, where smaller clusters are divided into even smaller ones, leading to larger distortion. Nevertheless, special attention should be paid to the value of 1.41 from SOM with the transformer. In this model, the entire dataset is divided into two clusters, resulting in higher distortion. However, when comparing distortion with the same number of clusters, their distribution can be compared. For example, in SOM, the number of clusters for MLP, decision tree, and AutoEncoder calibrated data is 5. The decreasing distortion indicates a more even distribution of datasets. Similarly, for K-means, MLP-calibrated data demonstrate higher performance, while decision tree-calibrated data outperforms transformer-calibrated data in spectral clustering.

Table 4.6: The cluster distribution standard deviation for each combination from this project

algorithms	Transformer	MLP	Decision Tree	AutoEncoder
K-means	14.66	9.83	9.86	15.9
Spectral	20.23	14.88	16.51	19.31
SOM	1.41	35.51	22.91	17.58

In addition to the distribution of clusters, silhouette scores (displayed in Table 4.7) also need to be taken into consideration. It is evident that SOM with AutoEncoder achieves a significantly higher score compared to the others, indicating its superior performance. This finding aligns with the previous analysis of cluster distribution. Furthermore, when comparing different numbers of clusters, it is apparent that SOM with AutoEncoder consistently outperforms silhouette scores of 0.68, 0.25, 0.14, 0.10, and 0.12, which are the best scores achieved with 2, 4, 5, 6, and 8 clusters, respectively. This serves as evidence that SOM with AutoEncoder provides the best performance among all the combinations.

Table 4.7: The silhouettes score for each combination from this project

algorithms	Transformer	MLP	Decision Tree	AutoEncoder
K-means	0.15	0.14	0.12	0.44
Spectral	0.15	0.12	0.10	0.25
SOM	0.68	0.43	0.28	0.73

CHAPTER

FIVE

DISCUSSION

5.1 System validation

Upon obtaining the generated results, we sought consultation from experts in the field of water quality monitoring to gain insights and validate our findings.

To facilitate effective visualization and analysis, two pages were developed, as depicted in Figure 5.1. The first page displays historical data, while the second page presents real-time data. Each page consists of two main sections: filters on the left side, which enable the selection of specific sensor platforms, and visualization tools on the right side. Notably, the data analysis page (Figure 5.2) has three individually scrollable columns, allowing users to compare the performance of different sensors and their relationships. Users have acknowledged the value of these pages and acknowledged the enhancement they bring to the quality of collected sensor data.

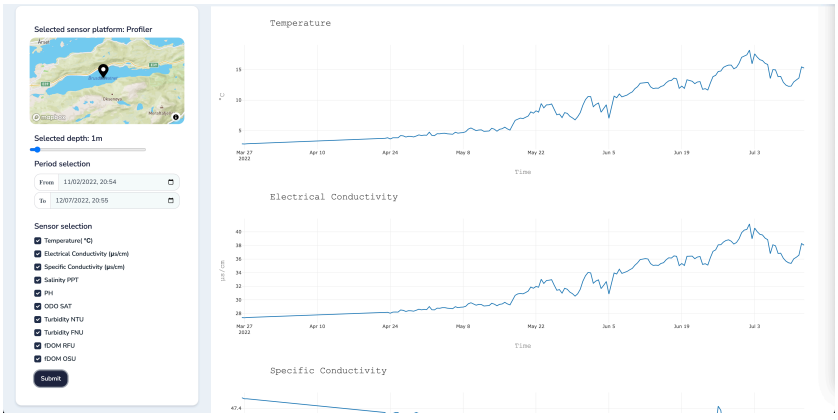


Figure 5.1: Historical data visualization page

During discussions on clustering results, the experts emphasized the significance of determining the appropriate number of clusters. In practice, it is uncommon to have more than five clusters, especially in the context of drinking water reservoirs where careful selection and minimal drastic changes are expected. However, merely having two clusters does not provide accurate enough results to represent the final outcome. Consequently, the experts recommended that three, four,

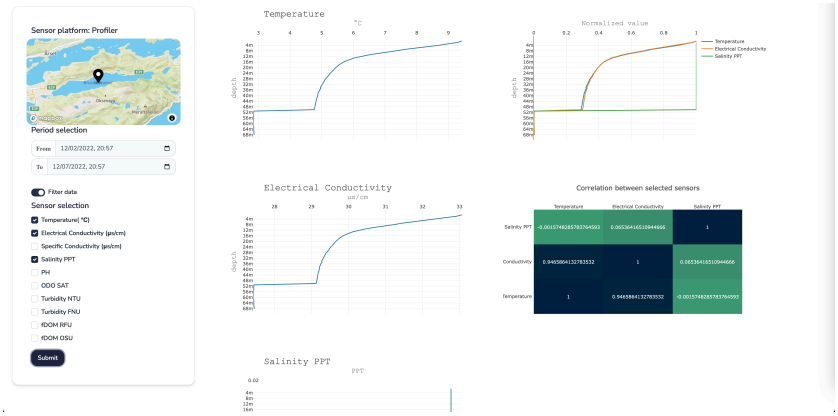


Figure 5.2: Data analysis page

or five clusters would be the most suitable choices. Subsequently, they examined the clustering results and confirmed that the clustering achieved using the Self-Organizing Map (SOM) with AutoEncoder calibrated data was reasonable and could be further explored through in-depth analysis.

The significance of this project in the Ålesund water quality monitoring system was underscored by the experts. They emphasized the system's seamless data analysis capabilities, which ultimately yield substantial benefits for stakeholders by providing valuable insights into potential fluctuations in water quality. To further illustrate the potential applications of the collected data, one example is the utilization of fluid dynamics simulations. These simulations can be performed using the collected data, allowing for the prediction and assessment of water body behavior and characteristics [69]. Moreover, the recorded data can be employed for water risk management purposes. The findings and observations derived from the monitoring system can serve as essential inputs for assessing and mitigating potential risks associated with water quality in the urban water supply system [70].

5.2 Potential limitaions

The approach outlined in this paragraph faces several limitations, specifically pertaining to environmental factors, workforce issues, and standardization challenges.

One of the primary limitations is related to the environmental conditions in which the experiments are conducted. The experimental platform is situated in a lake in Ålesund, and during the winter season, the platform freezes, rendering it incapable of collecting data. Consequently, this poses a significant obstacle in obtaining accurate and reliable data during those periods. Moreover, the surrounding environment further complicates data collection efforts. The reservoir is not located in an isolated area, as there is nearby transportation infrastructure that may introduce unwanted influences on the water quality sensors.

The second limitation revolves around the workforce involved in the project, particularly in terms of data labeling. Labeling the data is crucial not only for anomaly detection but also for enhancing calibration and clustering outcomes. Allocating additional resources to ensure the collection of accurate sensor data and establishing standardized clustering procedures would greatly improve the overall accuracy of the solution. However, acquiring access to water quality

data, which is confidential, proves to be challenging, hindering the expansion of the dataset. Consequently, human labeling remains the most viable option, despite requiring a substantial labor force.

The final limitation concerns the usability of the overall solution and the need for standardization. Establishing a comprehensive set of standards is essential to enhance the system's utility and comprehensibility. These standards would encompass indicator selection, data collection and processing procedures, as well as the final clustering methodology. However, each city follows its own unique set of standards, and different systems are employed, further complicating the task of standardization. Overcoming this challenge presents a difficult undertaking due to the inherent variations across different locations and systems.

5.3 Additional contribution

In addition to the research work presented in this thesis, I actively participated in various research projects during my master's studies at NTNU Ålesund. These projects encompassed a range of topics and allowed me to broaden my knowledge and gain valuable research experience:

Project list

Project	Gunnerus for the ship digital twin creation (Nov.2021-Jun.2022)
Description	This project aims to commercialize a ship digital twin for performance analysis and prediction based on the experimental vessel Gunnerus.
Responsibility	1. Building and maintaining the backend server with Python Django. 2. Manage the PostgreSql database. 3. Build the digital twin with Three.JS.
Results	1. Build the database and the backend server. 2. Implement the playback function to display historical data stored in the database. 3. Implement the real-time displaying function to show real-time data sent by the MQTT broker.

Project	PlastOPol for marine plastic litter monitoring (Nov.2021-Jun.2022)
Description	This project is intended to use machine learning to help identify the category of the litter. But as there is no applicable dataset for training, an App is needed for collecting data.
Responsibility	1. Design the interface of the app. 2. Build a database on a Linux server to manage the pictures and annotations uploaded. 3. Implement the App and conduct user studies.
Results	1. Build a database on a Linux server. 2. The app can upload pictures and relevant information to the server. 3. Users could make notation within the App. 4. Collected data can be used for retraining the machine learning model.

Project	Digital twin for repositioning turbines in a wind farm (Mar.2022 - Jun.2022)
Description	This project is intended to create a digital twin for analysis the influence of the location of wind turbines to the reduction of wake effect.
Responsibility	1. Create the digital twin which can interact with users to visualize the wake effect. 2. Implement the machine learning algorithm for positioning in Unity. 3. Academic writing.
Results	1. Build a digital twin of the wind farm that can provide suggestions on how to locate the wind turbines in different conditions. 2. Published a conference paper.

During the previous work, Two conference paper was accepted by the submitting date:

- Digital Twin-Driven Dynamic Repositioning of Floating Offshore Wind Farms
  Accepted by ICREC 2022
- PlastOPol: A Collaborative Data-driven Solution for Marine Litter Detection and Monitoring
  Accepted by ICIT 2023

CHAPTER

SIX

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The current water supply system relies on biological indicators for drinking water monitoring and heavy metal measurements for water source monitoring in Ålesund. However, these methods are evaluated over longer periods and need in-laborotary processing, making real-time monitoring impractical. To address this limitation, a sensor platform has been implemented to collect data from Brusdalsvatnet Lake in Ålesund. Additionally, an architecture for sensor data analysis has been proposed in this work. By implementing this proposed architecture, the water supply system can promptly identify anomalies, calibrate the data, and conduct clustering, ensuring effective management of water resources which contributes to advancements in water quality monitoring, providing valuable insights for stakeholders and decision-makers involved.

In this study, we employ a CNN-based anomaly detection method to identify and address anomalies present in the collected data, utilizing recurrence maps to encode the data. Furthermore, we evaluate the effectiveness of four different calibration methods for handling data with anomalies. These methods include the transformer method for analyzing time dynamics, MLP for exploring entity correlations, and decision tree and the proposed AutoEncoder-based networks for comprehensive CTS analysis. Through the application of these calibration techniques, a dataset of higher quality is generated, enhancing the reliability and accuracy of subsequent analyses. To facilitate clustering and categorization of the processed data, we encode the data into 2D tensors. These tensors are then fed into clustering algorithms, such as K-means, spectral clustering, and SOM. This clustering process allows for the creation of distinct categories representing different water quality levels. By applying these clustering algorithms, a comprehensive understanding of the water quality situation in Brusdalsvatnet Lake can be obtained.

Based on the analysis of the results, it is noted that the SOM cluster with AutoEncoder calibrated data outperform the other combinations with a higher silhouette score which indicates a smaller in-cluster distance and larger intra-cluster distance. This finding provides evidence that both the time dynamics and synchrony among the sensors and depths play a crucial role in analyzing water quality data. When calibrating the data using AutoEncoder, the data is first encoded into a tensor where the time series are grouped together considering the synchrony among different

sensors. Additionally, when clustering the data using SOM, the grouping is based on spatial location, taking into account the correlation among different entities to capture the temporal changes in water quality levels. Hence, the conclusion can be drawn that the water quality data analysis can be done with a CTS analysis method.

Regarding the first research question on the architecture design for automatic water quality data collection, processing, and clustering, it is addressed through a comprehensive data-driven solution utilizing a pre-built sensor platform. The sensor platform located at Brusdalsvatnet Lake collects data and wirelessly transfers them to a local server, where the proposed model in this thesis is employed for anomaly detection, signal calibration, and data clustering. Throughout the entire process, except for labeling the training data for anomaly detection, no human labeling is required. This approach significantly reduces the need for manual intervention, allowing for close monitoring of water quality in this drinking water source.

For the second research question concerning signal enhancement, it is achieved by applying a CNN-based anomaly detection method and CTS data calibration. For anomaly detection, a CNN network is employed that achieves an accuracy of 99.6% with the test data. Subsequently, the anomalies are calibrated using four algorithms in which the AutoEncoder algorithm regenerates data that more closely resembles the original clean data making it the preferred calibration algorithm. This choice is based on the fact that the water quality does not vary significantly during the testing period, as indicated by the records.

Addressing the third research question on the water quality categories, clustering algorithms are utilized to group the data into smaller clusters. Particularly, the SOM model achieves a high silhouette score of 0.73 when the signal data is calibrated with AutoEncoder. Once new data is generated, it can be fed into this model, automatically classifying the data into distinct water quality levels represented by different clusters.

When evaluating the impact of this system on the overall water source management system, this platform and data-driven solution can be integrated with a more comprehensive Cyber Physical System. The collected data offers valuable insights into the current state of water quality. By combining them with the capabilities of the CTS, various important applications become feasible. These include risk management, fluid dynamic simulation, and accurate water quality record-keeping. The outcomes of these advanced analyses serve as valuable references for stakeholders and decision-makers involved in water source management.

In summary, this project successfully addresses the four research questions. However, certain limitations exist that constrain the performance of the project.

Firstly, the training of the model requires a substantial amount of data to improve its performance. However, in Norway, water quality data is considered sensitive and access is restricted without permission. This limitation restricts our data source, necessitating more time to generate data from our platform. Even with sufficient data, the quality of the data is crucial. For example, the best-performing calibration model, AutoEncoder, is unable to produce a distribution of calibrated data that exactly matches the clean data. This discrepancy arises due to the nearly 1:1 ratio between clean data and data that requires calibration, making it challenging for the model to thoroughly learn the underlying patterns and ultimately affecting its performance.

Secondly, the model is tailored to the current system, and if applied to other projects with different indicators or spatial relationships, the model would require retraining, consuming both time and data resources. Additionally, as the dataset grows larger, computational power becomes

a requirement for efficient model execution.

The third limitation concerns the explanation of the results obtained. Presently, the results are solely generated from the perspective of data scientists, lacking any domain-specific explanations. This aspect diminishes the practicality of the model in real-life scenarios and limits its reliability for external users.

6.2 Future work

For future work, this project can be extended into a more comprehensive and large-scale endeavor, with the potential to be implemented in real-life scenarios and provide valuable information to various users, including decision-makers in urban facility design and citizen scientists. However, in order to achieve this, it is imperative to enhance the reliability of our model.

Firstly, to improve the performance of our model in anomaly detection, data calibration, and clustering, it is necessary to gather a larger volume of data. Currently, data collection is limited to a single fixed sensor platform. Expanding the project to incorporate additional platforms and floating platforms would offer a more comprehensive perspective on the overall water quality of the reservoir. Furthermore, increasing the sampling frequency would enable more real-time data acquisition, enhancing the timeliness and accuracy of our model.

Secondly, seeking the guidance and expertise of professionals in the field of water quality monitoring would greatly enhance the meaningfulness and interpretability of our clustering results. Collaborating with domain experts would allow us to validate the clustering outcomes and gain insights into the underlying significance of these clusters over time. By establishing a concrete understanding of the implications of water quality changes, we can provide more precise and relevant information, thereby improving the practicality and utility of our project.

The third area of future work involves integrating our system with other existing systems. As mentioned in the introduction, the sensor platform can be considered a subsystem of a comprehensive CPS. This CPS may encompass various systems such as water treatment, water supply, or industrial production, where the source and outlet of water play a critical role. By incorporating our platform into these larger systems, stakeholders can gain real-time monitoring capabilities and insights into the impact of different activities on the drinking water source. This integration would provide a comprehensive and holistic view of the environment and drinking water quality, enabling proactive decision-making and ensuring the sustainability and safety of water-related processes.

In summary, the future work of this project involves expanding its scope, collecting more data, collaborating with domain experts to validate and interpret the results, and integrating the system with other relevant systems. By pursuing these avenues, we can improve the reliability, usability, and applicability of our model, enabling its effective implementation in real-life scenarios and benefiting various stakeholders involved in water quality management and decision-making processes.

REFERENCES

[1] James I. Price and Matthew T. Heberling. "The Effects of Source Water Quality on Drinking Water Treatment Costs: A Review and Synthesis of Empirical Literature". In: Ecological Economics 151 (2018), pp. 195–209. issn: 0921-8009. doi: https://doi.org/10.1016/j.ecolecon.2018.04.014. url: https://www.sciencedirect.com/science/article/pii/S0921800917316464.

[2] Rachna Bhateria and Disha Jain. "Water quality assessment of lake water: a review". In: Sustainable Water Resources Management 2.2 (2016), pp. 161–173.

[3] SC Lahiry. "Impact on the environment due to industrial development in Chhattisgarh region of Madhya Pradesh". In: Finance India 10.1 (1996), pp. 133–136.

[4] Prabhat K. Singh and Sonali Saxena. "Towards developing a river health index". In: Ecological Indicators 85 (2018), pp. 999–1011. issn: 1470-160X. doi: https://doi.org/10.1016/j.ecolind.2017.11.059. url: https://www.sciencedirect.com/science/article/pii/S1470160X17307719.

[5] Muinul H Banna et al. "Online drinking water quality monitoring: review on available and emerging technologies". In: Critical Reviews in Environmental Science and Technology 44.12 (2014), pp. 1370–1421.

[6] Pemysl Soldán. "Improvement of online monitoring of drinking water quality for the city of Prague and the surrounding areas". In: Environmental Monitoring and Assessment 193.11 (2021), pp. 1–12.

[7] Grzegorz Pasternak, John Greenman, and Ioannis Ieropoulos. "Self-powered, autonomous Biological Oxygen Demand biosensor for online water quality monitoring". In: Sensors and Actuators B: Chemical 244 (2017), pp. 815–822. issn: 0925-4005. doi: https://doi.org/10.1016/j.snb.2017.01.019. url: https://www.sciencedirect.com/science/article/pii/S0925400517300199.

[8] Zhining Shi et al. "Alternative particle compensation techniques for online water quality monitoring using UV–Vis spectrophotometer". In: Chemometrics and Intelligent Laboratory Systems 204 (2020), p. 104074.

[9] Manoj Kumawat et al. "Occurrence and seasonal disparity of emerging endocrine disrupting chemicals in a drinking water supply system and associated health risk". In: Scientific Reports 12.1 (2022), p. 9252.

[10] Marco Carminati et al. "A self-powered wireless water quality sensing network enabling smart monitoring of biological and chemical stability in supply systems". In: Sensors 20.4 (2020), p. 1125.

[11] María Custodio and Richard Peñaloza. "Data on the spatial and temporal variability of physical-chemical water quality indicators of the Cunas River, Peru". In: Chemical Data Collections 33 (2021), p. 100672. issn: 2405-8300. doi: https://doi.org/10.1016/j.cdc.2021.100672. url: https://www.sciencedirect.com/science/article/pii/S2405830021000264.

[12] Huzein Fahmi bin Hawari, Mohamad Nor Syahid bin Mokhtar, and Sohail Sarang. "Development of Real-Time Internet of Things (IoT) Based Water Quality Monitoring System". In: International Conference on Artificial Intelligence for Smart Community. Springer. 2022, pp. 443–454.

[13] Alessio Fascista. "Toward integrated large-scale environmental monitoring using WSN/UAV/Crowdsensing: a review of applications, signal processing, and future perspectives". In: Sensors 22.5 (2022), p. 1824.

[14] Lehlogonolo Ledwaba and HS Venter. "A threat-vulnerability based risk analysis model for cyber physical system security". In: (2017).

[15] Tanya Garg, Surbhi Khullar, and Gurjinder Kaur. "Security Issues and Challenges for Cyber-Physical Systems". In: Cyber-Physical Systems. Chapman and Hall/CRC, 2022, pp. 161–176.

[16] Razvan-Gabriel Cirstea et al. "Correlated time series forecasting using multi-task deep neural networks". In: Proceedings of the 27th acm international conference on information and knowledge management. 2018, pp. 1527–1530.

[17] Jilin Hu et al. "Risk-aware path selection with time-varying, uncertain travel costs: a time series approach". In: The VLDB Journal 27 (2018), pp. 179–200.

[18] Hallvard Ødegaard, Østerhus Stein, and Esa Melin. A 209 Veiledning i mikrobiell barriere analyse (MBA). Tech. rep. 209/2014. Norsk Vann, 2021.

[19] Jianjun Liao et al. "Wireless water quality monitoring and spatial mapping with disposable whole-copper electrochemical sensors and a smartphone". In: Sensors and Actuators B: Chemical 306 (2020), p. 127557.

[20] Simitha K.M. and Subodh Raj M.S. "IoT and WSN Based Water Quality Monitoring System". In: 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA). 2019, pp. 205–210. doi: 10.1109/ICECA.2019.8821859.

[21] Yiheng Chen and Dawei Han. "Water quality monitoring in smart city: A pilot project". In: Automation in Construction 89 (2018), pp. 307–316.

[22] S.J. Burian et al. "5.06 - Climate Vulnerabilities and Adaptation of Urban Water Infrastructure Systems". In: Climate Vulnerability. Ed. by Roger A. Pielke. Oxford: Academic Press, 2013, pp. 87–107. isbn: 978-0-12-384704-1. doi: https://doi.org/10.1016/B978-0-12-384703-4.00509-8. url: https://www.sciencedirect.com/science/article/pii/B9780123847034005098.

[23] Abdulbasit Alaribi, Abbas Elazhari, and Omar A Zargelin. "PLC Based a Robust Solution for an Urban Area Water System Dilemma". In: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC). IEEE. 2021, pp. 1300–1305.

[24] Di Wu, Hao Wang, and Razak Seidu. “Toward A Sustainable Cyber-Physical System Architecture for Urban Water Supply System”. In: 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics). IEEE. 2020, pp. 482–489.

[25] Di Wu et al. “Quality risk analysis for sustainable smart water supply using data perception”. In: IEEE transactions on sustainable computing 5.3 (2019), pp. 377–388.

[26] Jun Ma et al. “Soft detection of 5-day BOD with sparse matrix in city harbor water using deep learning techniques”. In: Water research 170 (2020), p. 115350.

[27] Yirui Wu et al. “Attention neural network for water image classification under IoT environment”. In: Applied Sciences 10.3 (2020), p. 909.

[28] Mohsin Munir et al. “DeepAnT: A deep learning approach for unsupervised anomaly detection in time series”. In: Ieee Access 7 (2018), pp. 1991–2005.

[29] Dan Li et al. “MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks”. In: Artificial Neural Networks and Machine Learning–ICANN 2019: Text and Time Series: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part IV. Springer. 2019, pp. 703–716.

[30] Chuxu Zhang et al. “A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data”. In: Proceedings of the AAAI conference on artificial intelligence. Vol. 33. 01. 2019, pp. 1409–1416.

[31] Fatih S Bayram et al. “Anomaly detection of multivariate image time series based on Gramian angular field using convolutional autoencoder”. In: International Workshop on Automation, Control, and Communication Engineering (IWACCE 2022). Vol. 12492. SPIE. 2022, pp. 330–335.

[32] Paul Boniol and Themis Palpanas. “Series2graph: Graph-based subsequence anomaly detection for time series”. In: arXiv preprint arXiv:2207.12208 (2022).

[33] Ahmed Shoyeb Raihan and Imtiaz Ahmed. “A Bi-LSTM Autoencoder Framework for Anomaly Detection–A Case Study of a Wind Power Dataset”. In: arXiv preprint arXiv:2303.09703 (2023).

[34] Lawrence Mwenda Muriira, Zhiwei Zhao, and Geyong Min. “Exploiting linear support vector machine for correlation-based high dimensional data classification in wireless sensor networks”. In: Sensors 18.9 (2018), p. 2840.

[35] Andrea Borghesi et al. “Anomaly detection using autoencoders in high performance computing systems”. In: Proceedings of the AAAI Conference on artificial intelligence. Vol. 33. 01. 2019, pp. 9428–9433.

[36] Mikel Canizo et al. “Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study”. In: Neurocomputing 363 (2019), pp. 246–260.

[37] Brian Morris. “Explainable anomaly and intrusion detection intelligence for platform information technology using dimensionality reduction and ensemble learning”. In: 2019 IEEE AUTOTESTCON. IEEE. 2019, pp. 1–5.

[38] Umer Saeed et al. “Fault diagnosis based on extremely randomized trees in wireless sensor networks”. In: Reliability engineering & system safety 205 (2021), p. 107284.

[39] Fabio Carrara et al. “Deep learning for structural health monitoring: An application to heritage structures”. In: arXiv preprint arXiv:2211.10351 (2022).

[40] Naomi Zimmerman et al. “A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring”. In: Atmospheric Measurement Techniques 11.1 (2018), pp. 291–313.

[41] Hongwei Guo et al. “A machine learning-based strategy for estimating non-optically active water quality parameters using Sentinel-2 imagery”. In: International Journal of Remote Sensing 42.5 (2021), pp. 1841–1866.

[42] Senliang Bao et al. “Correction of satellite sea surface salinity products using ensemble learning method”. In: IEEE Access (2021).

[43] Carolina Tenjo et al. “A new algorithm for the retrieval of sun induced chlorophyll fluorescence of water bodies exploiting the detailed spectral shape of water-leaving radiance”. In: Remote Sensing 13.2 (2021), p. 329.

[44] Philipp M Maier, Sina Keller, and Stefan Hinz. “Deep learning with WASI simulation data for estimating chlorophyll a concentration of inland water bodies”. In: Remote Sensing 13.4 (2021), p. 718.

[45] Bowen Yu et al. “Global chlorophyll-a concentration estimation from moderate resolution imaging spectroradiometer using convolutional neural networks”. In: Journal of Applied Remote Sensing 14.3 (2020), pp. 034520–034520.

[46] Sabine Arnault et al. “A tropical Atlantic dynamics analysis by combining machine learning and satellite data”. In: Advances in Space Research 68.2 (2021), pp. 467–486.

[47] Xili Wang, Li Fu, and Chansheng He. “Applying support vector regression to water quality modelling by remote sensing data”. In: International journal of remote sensing 32.23 (2011), pp. 8615–8627.

[48] Jiang Yu et al. “Using machine learning to reveal spatiotemporal complexity and driving forces of water quality changes in Hong Kong marine water”. In: Journal of Hydrology 603 (2021), p. 126841. issn: 0022-1694. doi: https://doi.org/10.1016/j.jhydrol.2021.126841. url: https://www.sciencedirect.com/science/article/pii/S002216942100891X.

[49] Özer Çinar and Hasan Merdun. “Application of an unsupervised artificial neural network technique to multivariant surface water quality data”. In: Ecological research 24.1 (2009), pp. 163–173.

[50] Salam Hussein Ewaid et al. “Development and evaluation of a water quality index for the Iraqi rivers”. In: Hydrology 7.3 (2020), p. 67.

[51] Mansi Tripathi and Sunil Kumar Singal. “Use of principal component analysis for parameter selection for development of a novel water quality index: a case study of river Ganga India”. In: Ecological Indicators 96 (2019), pp. 430–436.

[52] Xingguo Chen et al. “Two novelty learning models developed based on deep cascade forest to address the environmental imbalanced issues: A case study of drinking water quality prediction”. In: Environmental Pollution 291 (2021), p. 118153. issn: 0269-7491. doi: https://doi.org/10.1016/j.envpol.2021.118153. url: https://www.sciencedirect.com/science/article/pii/S0269749121017358.

[53] Tianan Deng, Kwok-Wing Chau, and Huan-Feng Duan. “Machine learning based marine water quality prediction for coastal hydro-environment management”. In: Journal of Environmental Management 284 (2021), p. 112051.

[54] Runzi Wang, Jun-Hyun Kim, and Ming-Han Li. “Predicting stream water quality under different urban development pattern scenarios with an interpretable machine learning approach”. In: Science of The Total Environment 761 (2021), p. 144057.

[55] Yuanhong Li et al. “Lagoon water quality monitoring based on digital image analysis and machine learning estimators”. In: Water research 172 (2020), p. 115471.

[56] Umar Islambekov and Yulia R Gel. “Unsupervised space–time clustering using persistent homology”. In: Environmetrics 30.4 (2019), e2539.

[57] Ehsan Fathi, Rasool Zamani-Ahmadmahmoodi, and Rafat Zare-Bidaki. “Water quality evaluation using water quality index and multivariate methods, Beheshtabad River, Iran”. In: Applied Water Science 8.7 (2018), pp. 1–6.

[58] Mehdi Ahmadmoazzam, Yaser Tahmasebi Birgani, and Mohsen Molla-Norouzi. “Assessment of the Water Quality of Karun River Catchment Using Artificial Neural Networks-self-Organizing Maps and K-Means Algorithm”. In: Journal of Environmental Accounting and Management 9.01 (2021), pp. 43–58.

[59] Omolbani Mohammadrezapour, Ozgur Kisi, and Fariba Pourahmad. “Fuzzy c-means and K-means clustering with genetic algorithm for identification of homogeneous regions of groundwater quality”. In: Neural Computing and Applications 32.8 (2020), pp. 3763–3775.

[60] Ashish Vaswani et al. “Attention is all you need”. In: Advances in neural information processing systems 30 (2017).

[61] Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. “Tranad: Deep transformer networks for anomaly detection in multivariate time series data”. In: arXiv preprint arXiv:2201.07284 (2022).

[62] Prajit Ramachandran, Barret Zoph, and Quoc V Le. “Searching for activation functions”. In: arXiv preprint arXiv:1710.05941 (2017).

[63] Pascal Vincent et al. “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.” In: Journal of machine learning research 11.12 (2010).

[64] David Arthur and Sergei Vassilvitskii. “How slow is the k-means method?” In: Proceedings of the twenty-second annual symposium on Computational geometry. 2006, pp. 144–153.

[65] Andrew Ng, Michael Jordan, and Yair Weiss. “On spectral clustering: Analysis and an algorithm”. In: Advances in neural information processing systems 14 (2001).

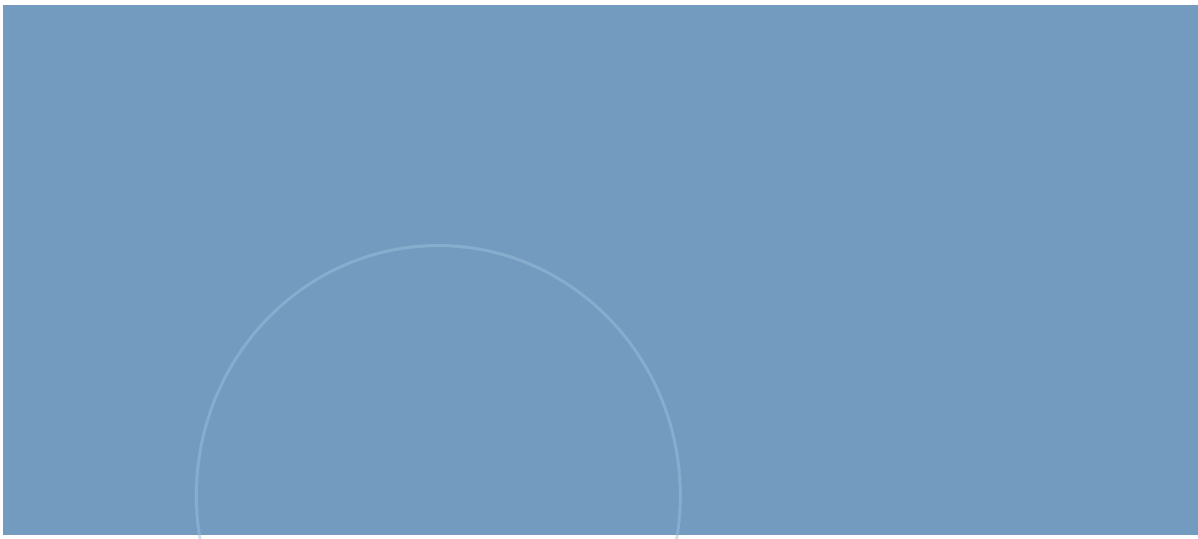
[66] Teuvo Kohonen. Self-organization and associative memory. Vol. 8. Springer Science & Business Media, 2012.

[67] Jincheng Liu. source code for the masters’ thesis. `https://github.com/Traversal2021/Master-Thesis-Source-Code`. Accessed: June, 2023. 2023.

[68] Ketan Rajshekhar Shahapure and Charles Nicholas. “Cluster quality analysis using silhouette score”. In: 2020 IEEE 7th international conference on data science and advanced analytics (DSAA). IEEE. 2020, pp. 747–748.

[69] F Abunde Neba et al. “Simulation of two-dimensional attainable regions and its application to model digester structures for maximum stability of anaerobic treatment process”. In: Water research 163 (2019), p. 114891.

[70] Di Wu et al. “A Case-Based Reasoning Solution for Urban Drinking Water Quality Control”. In: 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys). IEEE. 2021, pp. 2454–2459.



NTNU
Norwegian University of
Science and Technology