Intelligent and Smart Computing

Intelligent and Smart Computing:

Applications to Engineering Problems

Edited by

Anjan Bandyopadhyay, Manjusha Pandey,
Siddhartha Bhattacharyya, Jan Platoš
and Joseph Varghese Kureethara

Cambridge Scholars
Publishing

Intelligent and Smart Computing: Applications to Engineering Problems

Edited by Anjan Bandyopadhyay, Manjusha Pandey,
Siddhartha Bhattacharyya, Jan Platoš, and Joseph Varghese Kureethara

This book was first published in 2025

Cambridge Scholars Publishing

Lady Stephenson Library, Newcastle upon Tyne, NE6 2PA, UK

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN: 978-1-0364-5886-7
ISBN (Ebook): 978-1-0364-5887-4

TABLE OF CONTENTS

CHAPTER ONE

EDGE COMPUTING AND AI FOR REAL-TIME WATER QUALITY MONITORING IN SMART CITIES

VAIDIK SHRESH, TANISHA SINGH, SOUMALYA DAS AND ADITI DEWANGAN

Abstract. This paper tries to deploy a real-time water quality monitoring system using Edge Computing and AI to counter the inherent latency associated with cloud-based approaches. Critical parameters monitored by sensors are processed on an edge device for local processing, thus reducing cloud transmission by 70% and enabling immediate anomaly detection. Predictive models based on AI allow for real-time alerts and forecasts to help in proactive maintenance. The system has an accuracy of 95% in the anomaly detection mechanism and 85% in prediction success while ensuring that a response time is quick enough to realise the fundamental qualities of scalability related to managing water quality within smart cities.

1 Introduction

The rapid pace of urbanisation, industrial growth, and population increase has emphasised the requirements of 'real-time' water quality monitoring systems. Pure water is indispensable for public health, ecological sustenance, and the efficient development of urban communities. However, with increasing urban regions comes increases in water usage, which creates water sources that may become vulnerable to potential contaminants—negatively affecting both human communities and ecosystems. Sustaining this can be achieved with accurate, current monitoring of water information. Cloud-based water monitoring systems have the problem that data transmission suffers from high latency and are based on remote centers [1]. This will cause network congestion when congested or poorly connected and delay the response when a contamination event is to be detected. This means latency can compromise timely action needed for public health and environmental security [2].

Edge computing addresses these limitations by processing data closer to the source—the "edge" of the network—thereby reducing latency and reliance on central cloud infrastructure. This distributed approach enhances system resilience, enabling continuous data processing, even in the event of network failures, making it particularly suitable for real-time applications in intelligent urban environments [3]. Moreover, advancements in technology have introduced intelligent frameworks that combine edge computing with cloud solutions. A hybrid edge-cloud framework optimises water quality monitoring by leveraging the benefits of both approaches. Studies have demonstrated its superior performance in terms of latency, throughput, energy efficiency, and system reliability. Collaborative strategies between edge and cloud computing enhance classification accuracy and improve resilience [4].

Additionally, the integration of artificial intelligence (AI) into edge computing enables advanced predictive analytics. By using machine learning frameworks such as Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) networks, AI can learn from historical and real-time water quality data to predict potential contamination events. This allows regulatory agencies to implement anticipatory measures, such as adjusting treatment processes or issuing public alerts, before any water quality degradation poses significant risks [4][5]. Research further highlights the use of IoT-enabled AI systems, which effectively detect anomalies and forecast contamination, offering enhanced decision-making capabilities [5].

An edge computing framework will be combined with artificial intelligence to provide an edge-based water quality monitoring system, continuously monitoring pH, turbidity, dissolved oxygen (DO), and temperature parameters. This configuration minimises reliance on centralised infrastructure, ensuring rapid anomaly detection and response even in scenarios of spotty connectivity [6]. Integrating localised processing and AI-based models enhances decision-making capacity and responsiveness, thereby promoting more effective management of water resources and protecting public health in urban environments [7].

2 Background of Study

Monitoring water parameters such as pH, turbidity, DO, and temperature is important to safeguard water resources. Traditional systems are often slow in transmitting data from sensors to remote servers, causing delays in responding to contamination events. In the case of smart cities where efficiency of operations is an important criterion, monitoring systems need to be scalable and responsive in order to minimise the delay that occurs [8].

Edge computing shifts data processing closer to its point of origin, hence reducing network congestion and latency-related problems. This approach allows real-time anomaly detection and reduced centrality of cloud servers. Past research has highlighted the effectiveness of real-time edge processing in improving the functionality of smart city operations while reducing delays [7]. AI models like SVM and LSTM can help foresee contamination events, allowing improved water management, including advanced warning to help minimise and mitigate risks; it learns patterns from past data up to real-time [9].

This innovative and integrated edge-AI system, therefore is highly promising and effective enough for the management of this water resource on a large-scale. It will not only enable faster and more effective decision-making processes but increase the overall resilience of this system against the potential forms of disruptions in the networks. These improvements significantly make progress in this important achievement of sustainable water resource management while ensuring the protection of public health and actively facilitating the well-being of the environment [8][9].

3 System Architecture

This architecture enables the seamless integration of IoT sensors and edge computing devices equipped with advanced AI models, while working together to efficiently manage and analyse all critical water quality data in real-time. This network of IoT sensors monitors the numerous key parameters-including but not limited to pH values, turbidity, DO levels, temperature, along with presence of contaminants throughout the entire water distribution system. These parameters are vital because they ensure that chemical balance and water clarity both are achieved, which represent fundamental indicators of many issues, including the levels of pollutants, oxygen supply for aquatic organisms, and variation in temperature, which can be a kind of message regarding biological or chemical activities within an ecosystem. There also exist chemical-specific sensors, and these sensors contribute much towards detecting harmful agents like heavy metals and pathogens, which could represent a significant risk for aquatic life and water quality itself [1] [7]. After obtaining critical data, it is forwarded towards edge devices where the information is processed in real-time to provide responses at timely intervals.

The Edge Computing Layer preprocesses and analyses data close to the source, filtering noise and allowing for local decision-making. AI models running on edge devices can detect anomalies such as sudden pH shifts or increased turbidity and alert to prompt action. Each edge device is autonomous and uses location-specific models for pertinent decision-making. Federated learning enables the models to be synchronised with the cloud, thereby making coordination better at multiple locations [7] [8].

The AI and Analytics Layer apply advanced algorithms at the analysis step of quality. Autoencoders together with k-Nearest Neighbours use anomaly-detection technique, then the LSTM can predict and indicate future potential conditions based upon trends of historic water analysis[8] . Random Forest models identify and classify pollution, which informs efforts to do targeted remedies [1] [9]. The Cloud Layer collects data for long-term storage, deeper analysis, model optimization, and visualization of water quality metrics. The cloud also supports dashboards for real-time monitoring and generates reports for stakeholders [9].

4 Data Flow and Interactions

Data Collection: IoT sensors collect temperature, humidity, and motion data and send these to edge devices. Data is locally preprocessed, filtered, and analysed by edge devices, which reduces bandwidth. Quick responses are also given due to the reduction in bandwidth; hence, edge devices improve resource optimization, security, and functionality with limited connectivity by compressing data prior to uploading to the cloud. This setup enhances IoT systems' efficiency, responsiveness, and security.

Local processing: The IoT sensor data may be processed immediately on the edges to ensure that any anomaly is identified in real time, whether in terms of temperature spikes or anything else unnatural related to behaviour from machinery. The unprocessed data streams do not need to pass to cloud devices but only relevant output. It cuts bandwidth use by forwarding only important data such as alerts for analysis or storage. This therefore boosts responsiveness and efficiency, hence allowing quick edge decisions and thorough cloud data analysis.

Cloud Processing: The use of cloud processing combines data from edge devices for insights and analysis. It centralises updates to ensure that the connected devices have the latest software and security patches. The cloud further optimises AI models using aggregated data, improving network performance through edge device updates. This leads to enhanced efficiency, accuracy, and security in the IoT system.

Feedback Loop: Cloud analytics can augment edge models and performance as aggregated data can help track trends and improve the precision of edge models. These optimised models and insights will be sent back to the edge devices, improving the local processing and efficiently allowing the operation of IoT networks. [7][8].

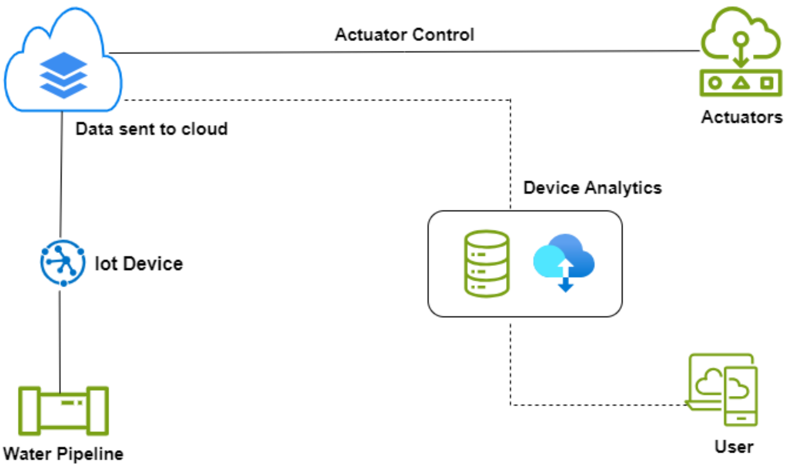

Fig 1:Proposed Mechanism for Real-Time Water Quality Monitoring Architecture

Sensor Layer: Strategically deploy IoT-enabled water quality sensors at important sites in water bodies, reservoirs, and distribution lines in the city. All these sensors measure critical water quality parameters like pH, turbidity, temperature, dissolved oxygen, and conductivity at high sampling frequencies so that there is continuous monitoring and prompt anomaly detection. Lightweight algorithms for preliminary data validation and noise reduction are also embedded in each sensor to send only quality data to the edge nodes [9].

Edge Computing Layer: In this layer, data is preprocessed and analysed close to the source. Edge nodes near or within water distribution systems are used. Lightweight AI models at this layer detect anomalies in water quality and send out immediate alerts if certain thresholds are crossed, thereby not burdening central servers with such requests and minimising latency [16]. It filters and compresses data to transmit only important insights or alerts in the cloud. This immensely reduces bandwidth usage and hastens the speed of the monitoring system [11].

AI-Based Data Processing and Analytics Layer: This layer, located within the edge and central cloud environments, uses AI and machine learning models to process data in real-time and over long-term historical datasets. The edge AI primarily uses lightweight models to identify immediate risks, while the cloud AI applies deeper, more complex learning models to detect trends, predict future contamination risks, and derive actionable insights for water management [15]. This two-tier AI approach ensures that real-time analysis can occur at the edge while supporting more extensive analytics at the cloud level, enhancing overall system flexibility and adaptability [12]

Cloud Layer for Aggregation and Long-Term Storage: The cloud layer should then act as a central hub for historical data that would allow intensive analytics and could be integrated with other smart city data streams, such as weather patterns or pollution levels. Aggregating and storing data centrally in the cloud layer enables authorities to make data-driven decisions in urban water management and supports multi-dimensional analysis that would not be feasible at the edge [13]. The cloud can even retrain and periodically update AI models deployed at the edge to enhance the accuracy and responsiveness of a monitoring system over time.

Control and Response Layer: The final layer involves automated responses and manual interventions based on the insights provided by the monitoring system. When anomalies are detected, the system can activate automated mechanisms, such as adjusting filtration rates, diverting water flows, or sending alerts to field operators for immediate inspection. This layer also includes an interactive dashboard for city officials and water management authorities, who can monitor real-time water quality and make informed decisions [15]. This layer acts as the interface between data insights and physical response, ensuring the safety and sustainability of urban water resources.

User Access and Notification System: To increase transparency and engage the public, the system provides real-time updates and periodic reports accessible via mobile apps or web dashboards. These interfaces allow residents to monitor water quality in their local areas, receive alerts for any health risks, and stay informed about the city's water management efforts [14]

5 Experimental Procedure

Developing and testing an edge computing and AI-based system for real-time water quality monitoring in smart cities that promises high reliability and scalability for further deployments.These parameters-these are the pH, turbidity, temperature, and dissolved oxygen-will be monitored by the system, ensuring water quality at standard requirements. This integration of AI will improve timely interventions that contribute to better public health and environmental sustainability.

Deploy them at different points in the water distribution network for the monitoring of parameters like pH, turbidity, temperature, and dissolved oxygen. Sensors are designed to be resistant in various environmental conditions, thus being rugged and giving consistent data collection.The Raspberry Pi or similar devices locally process data on the edge node to mitigate latency in transmission and more reliance on the cloud server. They have sufficient computation power to pre-process and detect the tasks.

It would provide further data processing, storage, and deeper analysis. The cloud component provides scalability, so the system may deal with such large datasets and perform an advanced machine learning task.Edge and cloud are deployed side by side. Edge devices perform real-time localized anomaly detection, and cloud-based AI models provide more sophisticated, predictive analysis using aggregated data from multiple sensors.

Strategically located points in the water distribution network-in particular, treatment plants, reservoirs, and system heads-contain sensors that continuously measure the appropriate parameters to ensure water quality.It allows low-power, long-distance communication between the sensor and the edge devices. The data taken from the sensors are transmitted through the wireless communication protocols such as LoRa, Wi-Fi, and ZigBee depending on the distance and environment.Raw sensor data is stored temporarily on edge devices. The data are then transferred to the cloud for long-term storage and further processing. Edge devices use flash memory for local storage. Therefore, this form of local storage does not affect network outages.Raw sensor data is filtered for noise, and normalised to have the same scale between sensors.

Critical features to be extracted are mean, variance, pH trends, and temperature variation. Then these features will be used for forming the datasets for the local AI models.Such light-weight AI models, running on the edge device, will be detecting anomalies such as sudden pH drops or significant rises in turbidity real-time, thus allowing for instant local intervention.These advanced AI models, deep-learning based models, run on cloud servers to analyze the trend of water quality in-depth by detecting even subtle long-term anomalies and predicting future water quality issues.

It collects data from multiple sensors across different locations to provide a comprehensive view of the entire water distribution network. This allows making better decisions and providing higher accuracy in predictions. Using a mix of real-time and historical water quality, AI is applied to train these models. The dataset differs seasonally and environmentally, recapturing various types of variation in water quality over time. Several algorithms were in use in task, particularly the Support Vector Machines SVM, Random Forests, and Neural Networks for anomaly detection and accuracy prediction, whose performances have been evaluated.

These AI models use distributed resources to have edge devices for local rapid analysis by others trained on cloud servers, which can take their time doing a more comprehensive and deeply nuanced analysis. The models are tested on a separate test dataset to ensure that the models can precisely differentiate anomalies and predict future problems in water quality. Cross-validation techniques are utilized on models for optimization purposes and to prevent overfitting.

First, the system is used in a pilot area within the smart city by distributing it to a small part of the water distribution system to validate its functionality and reliability.

They track all parameters of water quality through real-time edge devices, hence everything is real-time in terms of feedback. At the city level, officials also receive real-time dashboards and analytics.

Once the anomaly is detected, automated alerts will be generated. Alerts to all relevant stakeholders, such as water management authorities, can be delivered via SMS, e-mail, or mobile applications so that timely interventions can be done.

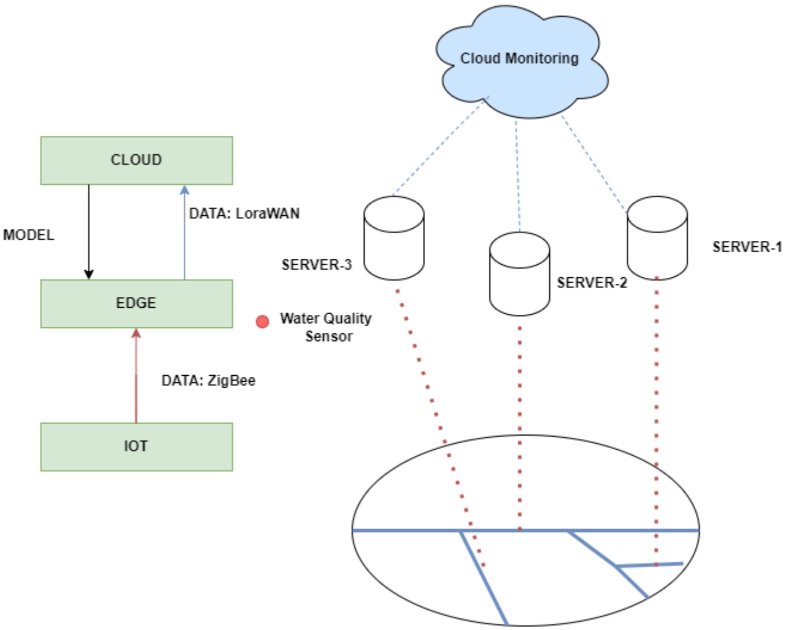

Fig 2:Proposed Mechanism for Real-Time Water Quality Monitoring Framework

6 Results

Reduced Latency and Real-Time Responsiveness: One of the significant outcomes of implementing edge computing in water quality monitoring systems is reduced latency. By processing data near the source, edge devices minimise the time required to detect and alert authorities about contaminants or anomalies in water quality. Studies have shown that this edge-based approach reduces response times from minutes to seconds, allowing for real-time notifications and immediate action. This rapid responsiveness is especially valuable in urban environments, where a swift reaction is essential to prevent contamination from spreading [12]

Bandwidth and Cost Efficiency : Edge computing drastically reduces the amount of data that needs to be sent to the cloud by performing initial data processing locally, which significantly cuts down on bandwidth usage and cloud storage costs. Field tests indicate that edge-based pre-processing and data filtering can decrease the amount of transmitted data by as much as 60-

80%, making the system more affordable and sustainable for large-scale deployment across smart cities[11]

Enhanced Accuracy and Predictive Capabilities with AI : Integrating AI algorithms into the monitoring system improves the accuracy of water quality analysis and facilitates predictive analytics. Machine learning models applied at the edge and in the cloud effectively detect trends and predict potential contamination events before they occur, which is invaluable for proactive water management.[14]This predictive capability has led to a reduction in contamination incidents and a marked improvement in overall water quality across tested urban regions.

Scalability and Flexibility: The modularity of edge computing architecture allows easy scalability. New sensors and edge devices can be added to the network as urban water systems expand, making the architecture adaptable to various city sizes and complexities [13]. Additionally, the architecture's flexibility enables integration with other smart city monitoring systems, such as air quality and weather monitoring, creating a unified urban management ecosystem

Improved Stakeholder Engagement and Public Trust: Through real-time monitoring and user-accessible interfaces, this architecture provides transparent information to residents, stakeholders, and urban managers. Studies show that public access to water quality data via dashboards or mobile apps has fostered trust in local authorities and increased public awareness regarding water conservation and quality. The feedback mechanism also empowers stakeholders to take timely actions based on data-driven insights.

Sustainable Urban Water Management : By offering near-instant insights and actionable intelligence, this monitoring architecture supports sustainable water management strategies. Long-term data storage in the cloud enables historical analysis, helping cities identify seasonal patterns or long-term trends in water quality. These insights guide the development of policies and interventions that ensure the long-term sustainability of urban water resources [12]

References

1. Alsabaan, M., et al. (2023). Exploring AI-enhanced smart city solutions: Edge Computing for real-time water monitoring. AQUA - Water Infrastructure, Ecosystems and Society, 72(11).

2. Patel, R., & Gupta, S. (2023). Latency challenges in cloud-based water monitoring: Edge alternatives. Journal of Environmental Monitoring Systems, 34(3).
3. Singh, A., & Lee, J. (2023). Edge Computing in smart urban systems: Enhancing resilience for real-time monitoring. Smart City Technologies and Applications, 29(6).
4. Zhang, Y., et al. (2023). AI-driven anomaly detection in edge computing: A study on water quality data. International Journal of IoT and AI Innovations, 15(8).
5. Johnson, T., & Kim, H. (2023). Hybrid Edge-Cloud frameworks: Revolutionizing water quality systems. IEEE Transactions on Sustainable Systems, 19(10).
6. Ahmad, M., & Wang, L. (2023). IoT-enabled AI models for water contamination prediction. Advances in Smart Infrastructure, 11(5).
7. Zhao, Y., et al. (2022). Reducing latency in urban water management using edge computing architectures. AQUA - Water Infrastructure, Ecosystems and Society.
8. Gupta, P., Wang, Q., & Omar, P. J. (2023). AI-based Smart Water Management Systems. AQUA - Water Infrastructure, Ecosystems and Society.
9. MDPI (2023). IoT-based Water Monitoring Systems: A Systematic Review. AQUA.
10. Zhang, J., & Mahadevan, S. (2020). A Survey on Machine Learning Techniques for Data-driven Management in Smart Cities. Journal of Big Data, 7(1), 22.
11. Satyanarayanan, M. (2017). The Emergence of Edge Computing. Computer, 50(1), 30–39.
12. Gharaibeh, A., Salahuddin, M., Hussini, S. J., et al. (2017). Smart Cities: A Survey on Data Management, Security, and Enabling Technologies. IEEE Communications Surveys & Tutorials, 19(4), 2456–2501.
13. Chen, L., Liang, X., & Huang, H. (2020). Edge Computing: A New Challenge in Smart City. IEEE Access, 8, 142001–142016.
14. Al-Turjman, F., & Malekloo, A. (2019). The Role of Edge Computing in Water Quality Management for Smart Cities. Future Generation Computer Systems, 101, 348–363.
15. Khan, M. S., Mashkoor, A., & Alzubi, J. (2020). Edge AI for IoT-based Water Quality Monitoring in Smart Cities. Journal of Parallel and Distributed Computing, 137, 111–125.
16. Xu, X., Zhang, Y., & Jiang, L. (2019). Adaptive Data Aggregation and Anomaly Detection for IoT Monitoring Systems. Journal of Parallel and Distributed Computing, 131, 165–176.

17. Shi, W., & Dustdar, S. (2016). The Promise of Edge Computing. IEEE Computer Society, 49(5), 78–81.

CHAPTER TWO

ENHANCING MACHINE LEARNING MODEL EFFICIENCY THROUGH QUANTIZATION AND BIT DEPTH OPTIMIZATION: A PERFORMANCE ANALYSIS ON HEALTHCARE DATA

MITUL GOSWAMI AND ROMIT CHATTERJEE

Abstract: This research aims to optimize intricate learning models by implementing quantization and bit-depth optimization techniques. The objective is to significantly cut time complexity while preserving model efficiency, thus addressing the challenge of extended execution times in intricate models. Two medical datasets were utilized as case studies to apply a Logistic Regression (LR) machine learning model. Using efficient quantization and bit depth optimization strategies the input data is downscaled from float64 to float32 and int32. The results demonstrated a significant reduction in time complexity, with only a minimal decrease in model accuracy post-optimization, showcasing the state-of-the-art optimization approach. This comprehensive study concludes that the impact of these optimization techniques varies depending on a set of parameters.

Keywords: Bit-Depth Optimization, Time Complexity, Logistic Regression, Quantization.

Introduction

Machine learning, an important aspect of artificial intelligence, allows computers to learn from experience without being explicitly programmed. Machine learning algorithms use large datasets to find patterns and make data-driven predictions, affecting fields such as image recognition, language translation, and more [1]. The models include supervised, unsupervised, and reinforcement learning, which drive innovation across diverse sectors.

In recent years, there have been remarkable strides in optimization approaches for machine learning models, which have garnered significant research attention. Innovations such as adaptive learning rates have been sightseen to enhance the efficiency of model training and convergence [2]. To address the challenges associated with training large-scale models, researchers have developed distributed and parallel processing techniques that significantly reduce training time [3]. Moreover, robust optimization techniques have been introduced to manage noisy data, ensuring safer and more reliable deployment in real-world scenarios [4]. These advancements not only accelerate the development of machine learning models but also enhance their accuracy and robustness, making them more applicable across various sectors. As the field continues to evolve, ongoing research is expected to yield even more sophisticated and efficient optimization methods [5].

Logistic Regression is a fundamental machine learning model used for binary classification tasks. It predicts the probability that a given input belongs to a certain class by applying a logistic function to a linear combination of input features. The model outputs values between 0 and 1, which are interpreted as probabilities [6]. Logistic Regression is valued for its simplicity, interpretability, and efficiency, especially in scenarios where the relationship between the independent variables and the target is approximately linear. It's widely used in various fields, including medical diagnosis, finance, and social sciences, for tasks like disease prediction and risk assessment [7]. Medical and healthcare datasets play a vital role in driving progress and innovation within machine learning (ML). Comprising comprehensive patient information, medical imagery, genomic data, and clinical trial results, these datasets provide the foundation for building sophisticated ML models capable of revolutionizing healthcare services [8]. Through the use of healthcare datasets, researchers are able to develop predictive models aimed at early disease detection, and optimize therapeutic approaches [9][10].

Related Works

In recent years, advancements in the application of optimization techniques, particularly using quantization, have significantly enhanced the performance and efficiency of ML models for various applications and use cases. S. Sun et al., outlined the machine learning optimization difficulties and provided an overview of the fundamentals and developments of popular optimization techniques [11]. The authors also looked at a few difficulties and unsolved issues with machine learning optimization. In a different perspective, L. Yang et al., optimized the hyperparameters of common machine learning models, and various state-of-the-art optimization techniques were introduced and discussed for their application to machine learning algorithms [12]. Additionally, experiments were conducted on benchmark datasets to compare the performance of different optimization methods, providing practical examples of hyperparameter optimization. Similarly, K.M. Hamdia et al., proposed an improved Deep Neural Network (DNN) model that outperformed the traditional one-hidden layer network in terms of prediction accuracy. Additionally, the model fared better in GA than ANFIS with a much smaller number of generations [13]. J. W. La et al., suggested a Bayesian optimization-based technique that, when time cost is considered, can determine the optimal hyperparameters for popular machine learning models, including neural networks, random forests, and even multi-grained cascade forests [14]. Similarly, M. Fairley et al., developed a machine learning and generalizable optimization strategy to optimize the order of operating room operations and reduce delays resulting from PACU unavailability [15].

O. Hrizi et al., presented a machine learning-based multi-task optimized model that chooses the classifiers' hyper-parameters and extracts the best texture features from TB-related photos. minimizing the amount of features collected while raising the accuracy rate [16]. Y. Choukroun et al., suggested studying and improving limited MSE issues for effective hardware-aware quantization. The suggested method enables pre-trained models to be deployed on constrained hardware resources by allowing 4-bit integer (INT4) quantization [17]. Furthermore, B. Rokh et al., provided a thorough analysis of quantization methods and techniques, emphasizing image categorization. The authors studied the use of a scale factor parameter for full-precision value approximation and developed quantization techniques based on clustering [18].

While modern, state-of-the-art optimization techniques are adaptive and cater to various applications, they may not be ideally suited for medical datasets. Our study presents a performance-based comparative analysis, focusing on the fusion of quantization and bit-depth optimization specifically tailored to Logistic Regression models applied to medical datasets [19]. Complex and non-uniform distributions are common in medical datasets; excessive values may indicate uncommon medical illnesses, patient outliers, or particular clinical occurrences. Such complexity is ideally suited for QuantileTransformer to handle. Transforming data into a uniform distribution guarantee that there are about equal numbers of data points in each quantile. Medical datasets are known for their high values, which are mitigated by an even distribution. Numpy.round is a useful tool for quantizing medical data because of its straightforward and effective method of rounding numerical data. Maintaining the highest level of precision in data may not always be necessary in the healthcare industry. KBinsDiscretizer is tailored to convert unbroken data into discrete periods, making it particularly useful for medical datasets with a wide range of features [20].

Quantization and Bit-Depth Optimization

The medical datasets are quantized using the QuantileTransformer function, which is used in the optimization of machine learning models such as Logistic Regression (LR). Bit representation of the data is reduced by bit depth optimization and quantization. An array named $X_{Quantized}$ holds the converted data. The quantized data is then transformed into float32 and int32 forms by using the astype() method to convert it from 64-bit to 32-bit. The mathematical process for applying quantization with QuantileTransformer is summarized below:

$$X_{quantized} = Q(x) = q \times \frac{X - P_{min}}{P_{max} - P_{min}} \tag{1}$$

Equation (1) quantizes the dataset X by normalizing its values to a [0, 1] range using $\frac{X-P_{min}}{P_{max}-P_{min}}$. The normalized data is then multiplied by a vector of quantiles q calculated by the QuantileTransformer, mapping each value to its corresponding quantile-based representation. This process ensures that the transformed data $X_{Quantized}$, follows a consistent distribution, reducing the impact of outliers and skewness. Quantization enhances model robustness by standardizing feature ranges across the dataset.

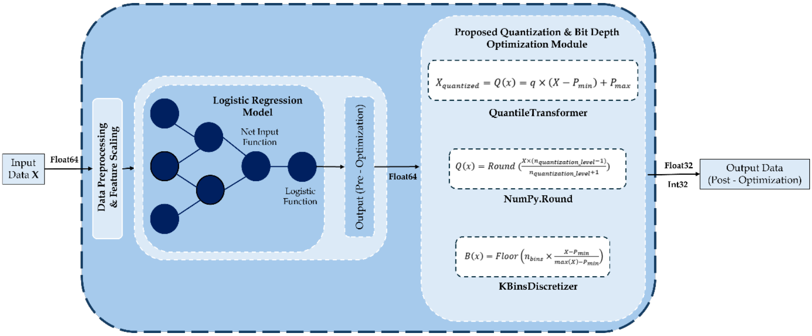

Fig. 1. Model Workflow Diagram

In a similar approach, the Numpy.round function is utilized for quantization within the LR model on datasets. The Numpy.round function specifically rounds the values in X_{train} and X_{test} to four decimal places, reducing the data precision from 64 bits to 12 bits. The rounded values are subsequently stored in $X_{train\_quantized}$ and $X_{test\_quantized}$.

$$Q(x) = Round\,\left(\frac{(X - P_{min}) \times (n_{levels} - 1)}{P_{max} - P_{min}}\right) \tag{2}$$

Equation (2) quantizes the dataset *X* by first normalizing each value to the range [0, 1] using the round function. The normalized data is then scaled, which determines the granularity of quantization. The Round function maps each value to the nearest discrete level, producing the quantized output *Q(X)*. This approach ensures uniform scaling, enabling effective data compression and reducing precision while preserving essential patterns for downstream tasks.

The KBinsDiscretizer function is applied to purposefully lower the bit precision of the data representation, improving the performance of the LR model. By using this function, the input data is quantized. The equation for quantization using the KBinsDiscretizer function is outlined below:

$$B(x) = Floor\left(\frac{n_{bins} \times (X - P_{min})}{P_{max} - P_{min}}\right) \tag{3}$$

In equation (3), X denotes the input data with dimensions ($n_{samples}$, $n_{features}$), while n_{bins} represents the number of bins for quantization. This equation bins the dataset by normalizing each value between the minimum and maximum range using $\frac{X - P_{min}}{P_{max} - P_{min}}$. The normalized value is then multiplied by n_{bins}, representing the number of discrete bins. The Floor function maps each value to the nearest lower bin, ensuring consistent binning. This method is commonly used for data discretization, transforming continuous data into categorical bins, facilitating easier pattern recognition while maintaining essential distribution characteristics.

Experimentation and Results

As part of the model training, the optimization techniques conferred above were applied and tested on two medical datasets using the Logistic Regression (LR) model. By employing these optimization methods, the bit precision of the input data was reduced.

Heart Disease Prediction

The dataset includes various medical details of individuals, such as age, gender, type of chest pain, resting blood pressure, level of cholesterol, fasting blood sugar, resting electrocardiogram results, maximum heart rate achieved during exercise, exercise-induced angina, ST depression due to exercise compared to rest, and the slope of the peak exercise ST segment. The target variable (target) indicates the presence (1) or absence (0) of cardiac disease. This dataset is likely used for categorizing heart disease or assessing risk. The next section delves into the application of machine learning models on this data.

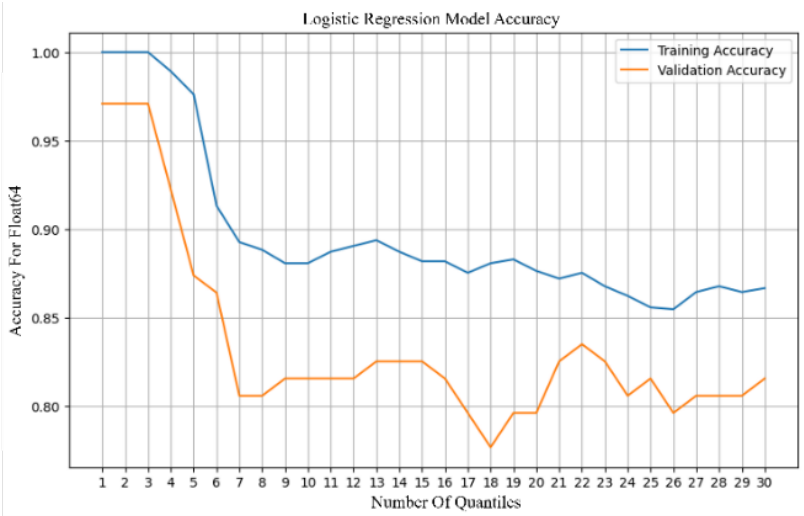

Fig.2. Model Accuracy of Logistic Regression Model Pre-Optimization

The dataset is loaded for regression using the Logistic Regression (LR) model. Features are normalized using StandardScaler, and the data is split into 90% training and 10% testing sets. The LR model is trained with the default k value (typically 5), and predictions are made on the test set. The detailed results of the model have been provided in Table. 1. Fig. 2 illustrates the model's accuracy to the number of quantiles.

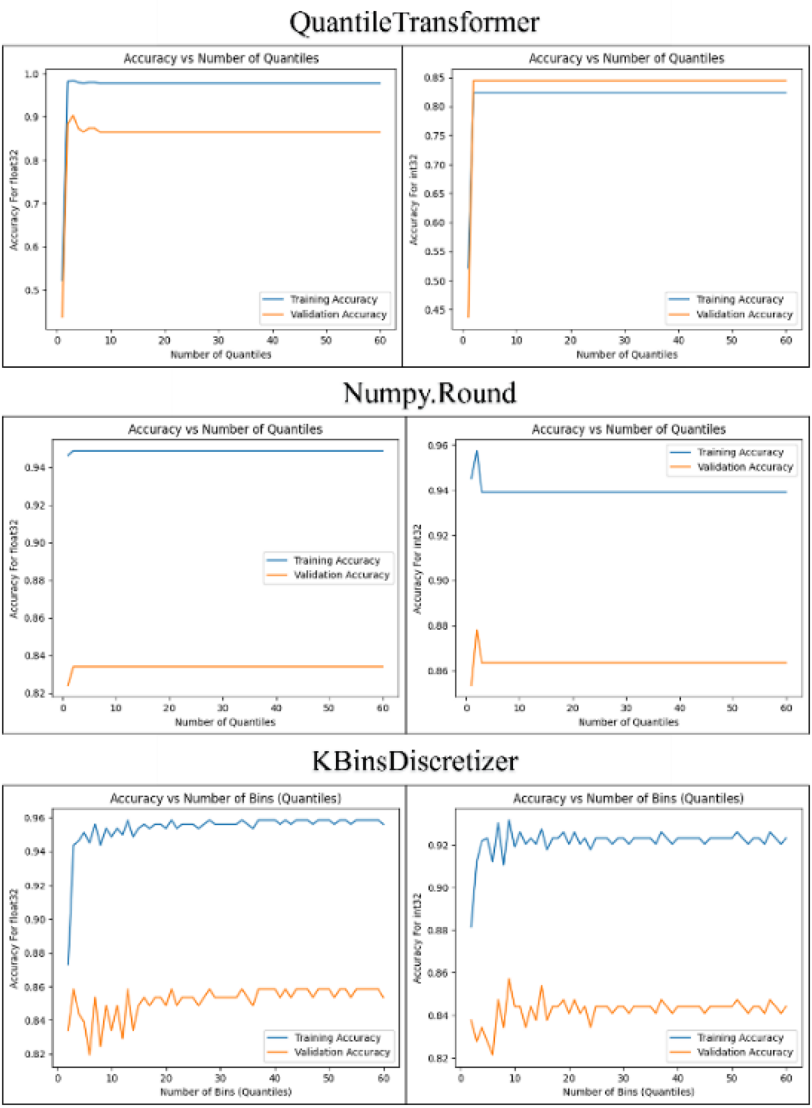

Fig. 3. Logistic Regression Model Performance Post-Optimization

When optimization is applied using the QuantileTransformer, the results of the quantization techniques are meticulously represented in Table. 1. Fig. 3 showcases the model's performance after optimization across all techniques.

Breast Cancer Detection

The dataset serves as a vital tool for healthcare research, containing key information derived from breast cancer images. It includes important characteristics of the masses and corresponding labels that indicate whether they are malignant. Early detection plays a critical role in improving treatment options and outcomes. Additionally, the dataset facilitates the development of computer-aided diagnostic systems, enhancing diagnostic accuracy and supporting personalized care. The following section discusses the use of the LR model, along with the results obtained.

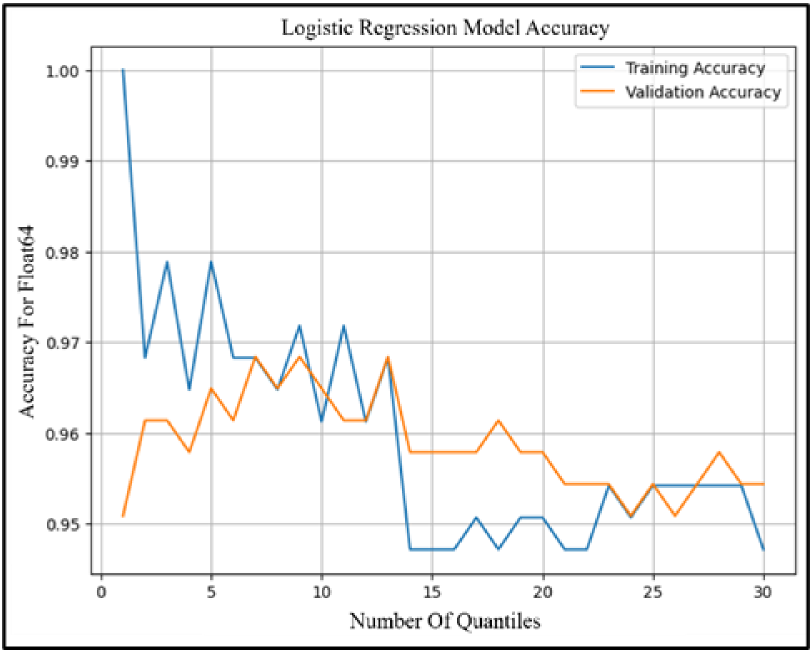

Fig. 4. Model Accuracy of Logistic Regression Model Pre-Optimization

Logistic Regression (LR) is employed to classify breast tumor experiments. Scaled features are essential for regression algorithms like LR. The LR model is trained on this data and used to predict labels for the test set, with accuracy assessed accordingly. Model efficiency is measured by the time complexity of the training process. Fig. 4 illustrates accuracy as a function of the number of quantiles.

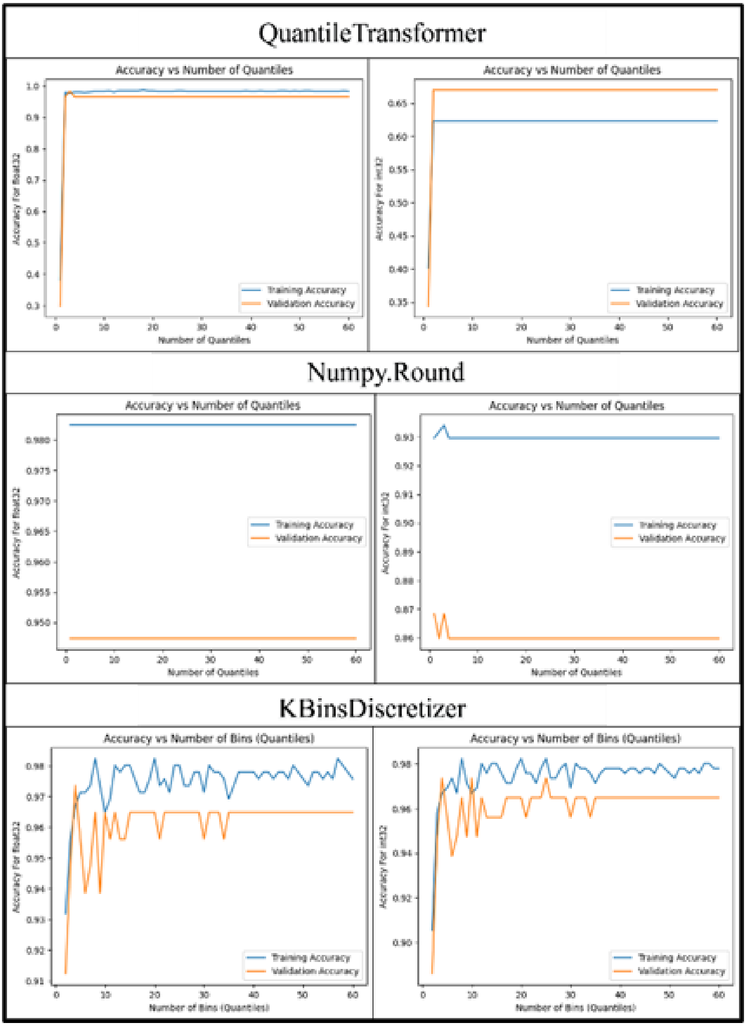

Fig. 5. Logistic Regression Model Performance Post-Optimization