2023-05

An IoT Intensive AI-integrated System for Optimized Surface Water Quality Profiling

Syeed, M M Mahbubul

Independent University, Bangladesh

Draft Version

An IoT Intensive AI-integrated System for Optimized Surface Water Quality Profiling

M M Mahbubul Syeed
Dept. of CSE & RIoT Center
Independent University, Bangladesh
Email: mahbubul.syeed@iub.edu.bd

Md. Rajaul Karim
RIoT Center
Independent University, Bangladesh
Email: rajaul@iub.edu.bd

Md Shakhawat Hossain
Dept. of CSE & RIoT Center
Independent University, Bangladesh
Email: shakhawat@iub.edu.bd

Kaniz Fatema
Dept. of CSE & RIoT Center
Independent University, Bangladesh
Email: kaniz.fatema@iub.edu.bd

Mohammad Faisal Uddin
Dept. of CSE & RIoT Center
Independent University, Bangladesh
Email: faisal@iub.edu.bd

Razib Hayat Khan
Dept. of CSE & RIoT Center
Independent University, Bangladesh
Email: rkhan@iub.edu.bd

***Abstract*—Surface water is heavily exposed to contamination as this is the ubiquitous source for the majority of water needs. This situation is exaggerated by excessive population, heavy industrialization, rapid urbanization, and ad-hoc monitoring. Comprehensive measurement and knowledge extraction of surface water pollution is therefore pivotal for ensuring safe and hygienic water use. However, current process of surface water quality profiling involves laboratory-based manual sample collection and testing, which is tardy, expensive, error-prone, and untraceable. This paper, therefore presents the design and development of an IoT integrated water quality profiling system that possesses a novel plug-and-play physical layer for the sensor actuation, and an AI powered fog computing based cloud application layer for remote water quality parameter measurement and data acquisition, remote data logging, monitoring and control, with data analytic for critical reasoning and decision making.**

I.I NTRODUCTION

Surface water is any body of water found on the Earth's surface, e.g., the freshwater in rivers, streams, lakes, wetlands, reservoirs, and creeks. This water is unequivocally the most indispensable natural element for the survival of mankind, and all living organisms, e,g., animals and plants [1], [2]. With rapid urbanization, change in livelihood, expansion of industrial and commercial complexes, surface water is becoming even more pivotal than ever before for maintaining the quality of life [1]. This water is the ubiquitous source for the majority of water needs, including drinking and domestic purposes, industrial and research activities, irrigation, horticulture, livestock farming and aquatic life including fish and fisheries [1], [3].

However, with this rapid growth of use, the surface water quality is increasingly declining due to heavy exposure to contamination and pollution [4]. Surface water can be contaminated in a number of ways, e.g., receiving industrial and domestic wastewater, sediments, and agricultural runoffs [1], [3]. Type and severity of contamination vary based on the establishment and its' water usage pattern. For instance, the water bodies close to heavy industrial zone are susceptible to heavy metals and hazardous substances that are discharged as by-product. The lands and water sources around the agricultural lands are exposed to fertilizers and residue of pesticides (e.g., organophosphate, carbamate) [1]. Alongside, the physical and chemical properties of the surface water vary considerably over time, e.g., sudden storm or flood can cause a gripping short-term change that may affect fisheries and pond water. Whereas, in long-term, water chemical properties vary with seasonal change and with the water usage pattern [3], [4].

Therefore, comprehensive measurement and knowledge extraction of surface water quality is an inevitable need to assess and predict the pollution, and derive factual observations to plan for sustainable use of the water resources, and associated disaster management [1]. However, investigation of the status quo reveals that the current process involves laboratory-based manual sample collection and testing which is tardy, time-consuming, expensive, ad-hoc, error-prone, and untraceable [1], [2]. Consequently, this process hinders timely assessment, decision making and long term planning for water quality assurance. Therefore, this paper empirically investigates and addresses the following set of research questions:

RQ1. What are the requirements for the development of a technology intensive pragmatic system for surface water quality profiling and management?

RQ2. How to glen the IoT principles and AI methods in designing an intelligent system for remote sensing, profiling and management of the same?

RQ3. How the system can be materialized by adopting cutting-edge technologies and methods?

II. RESEARCH APPROACH

Conversant with the research questions, this study adopts the *Constructive Research Approach (CRA)* as the core research method. By definition, CRA offers a methodology that creates innovative constructions to solve real world problems and contributes to the field of study where it is applied [2], [6]. Therefore, this method is exploited in achieving the following, (a) apprehend the pragmatic needs towards a knowledge-driven, integrated smart automation for water quality profiling, (b) design and develop the same by leveraging the best practised IoT technologies and design principles, AI-driven deep learning methods for analysis, decision making and remote sensing, and (c) document the validation of the system design and its' technical soundness.

III. The Conceptualization of the System Requirements

The exclusive summary of the system requirements for the Water Quality Management System (WQMS) are presented in Figure 1. In this figure a mapping among the system requirements and their realization within this system architecture is detailed. The cited requirements are ubiquitous, and require pervasive connectivity and computing ability with a consistent interaction between the physical and digital world using a plethora of sensors, actuators and communication network. Therefore the system is designed by adopting the *Internet of Things (abbreviated as IoT)* principles, design patterns and associated technologies [2], [3], [4]. IoT facilities an overarching system design utilizing the layered and client-server architectural design pattern [2]. The use of layered architecture ensures modular system design with clear separation of concept, reducing the dependency among system components, minimizing the inter-connectivity with higher degree of fault tolerance and maintenance [1], [2]. On the other hand, the client-server architecture powered by fog-computing assists a flexible network structure design with a secured narrow-band data communication channel among remotely connected devices [1], [2], [5].

A *Three-Layer* simplified reference stack architectural model is chosen to design the WQMS system, e.g., the Perception Layer, the Network and Data Process Layer, and the Application Layer. The abstract representation of the system architecture with their functional components and inter-connectivity is detailed in Figure 1.

The **Perception layer / Physical Layer** consist of 4 (four) inter-connected components with explicit functionalities to adequately serve the four core system requirements (as listed in Figure 1). These components are, the *Data Actuator, GNSS Module, Controller and Storage unit, and Monitor and Control unit*. The data actuator module is a custom build plug-and-play component that supports measuring of water quality parameters using digital sensors. The detailed H/W design specification for this module is reported in Section IV-A. The GNSS module records the localization information with the measured parameters, thus supporting the geo-tagged parameter measurement need. The monitor and control unit is the software module that provides real-time visualization of the actuated data, and allows monitoring and configuration of the sensors and associated components. Finally, the controller and storage unit implements the program logic required to offer real-time communication, control and temporary storage of sensor data which are periodically logged into the cloud database over the network layer. The detailed design specification of this layer can be found in Section IV-A.

The **Network and Data Process Layer / Transport layer** is the internet gateway between the perception and application layers as shown in Figure 1. This layer is responsible for the transmission and processing of information received to/from the perception layer and application layer. Therefore, effectively supporting the remote data logging, remote sensing and supervision services. A detailed technical design specification for this layer is documented in Section IV-B.

The **Application Layer** performs the raw data logging, data processing and storage with specialized services (e.g., intelligent decision support, AI driven autonomous data processing and visualization, exposure of API to access the data) and functionalities (e.g., user applications for remote monitoring, supervision, controlling and research) defined for different user groups. The application layer shown in Figure 1 is fully equipped with the necessary modules to serve these needs. The implementation detail of this layer is presented in Section IV-C.

IV. Technical Specification and Implementation of the WQSM System

A. Perception Layer

The perception layer of the **WQMS** system is consists of four independent yet interconnected modules, e.g., *the Data Actuator, Analog-to-digital Converter (ADC), the Micro-Controller & Storage (RaspberryPi), the GNSS & Cellular module, and the Monitor and control unit*. Figure 2 details the H/W specifications for each of the module along with their inter-connectivity.

The **Data Actuator** module is one of the core contributions of this work that consists of a modular, plug-and-play and extensible dynamic sensor hub for real-time extension of the required sensors. As shown in Figure 2, the data actuator is consists of a custom build *Dynamic sensor hub* and a *16 Channel multiplexer*. In this hub, up to 16 different sensors can be plugged and powered simultaneously. The multiplexer facilitates the 16 (C0-C15) input selection by 4 control signals (S0-S3) that are directly connected to GPIO 5, 6, 13 and 19 of the RaspberryPi micro-controller. This arrangement supports simultaneous polling of 16 sensors by the micro-controller. This design is extensible with additional attachment of multiplexers, and sensor hubs as per requirement.

The **ADC (Analog-to-Digital Converter) Module** converts the analogue sensor data (actuated by the sensors) into its' digital equivalent and transfer it to the micro-controller as shown in Figure 2. This is required as RaspberryPI micro-controller does not possesses any analogue input pin.

The **Micro-Controller & Storage** module is the processing unit which inter-connects and controls every other module in the Perception Layer. Several H/W alternatives are available for the implementation of this module. For instance, the Node MCU, Arduino, and Raspberry Pi. For this system development, the *RaspberryPi 4* integrated with the WiFi, Ethernet, SD card Slot, HDMI, Micro-USB Power Supply is used. Because, the system requirements (e.g., the clock speed, memory management, diverse interfacing alternatives and low power consumption) best matches with this micro-controller. The **Raspberry Pi Control Logic** is the system kernel for the Perception Layer and is burnt into the micro-controllers' memory. This kernel defines the control logic for the management of every system resource (e.g., the sensors, GNSS Module, SD card, User Interface, network connectivity and communication) to ensure proper functionality of this layer. The kernel also ensures optimized and authenticated resource access, data validation, storage management, and communication over the Network Layer or with the *Monitor and control unit*. The development language for the kernel is specific to the micro-controller in use, and in this case, it is the *Python*.

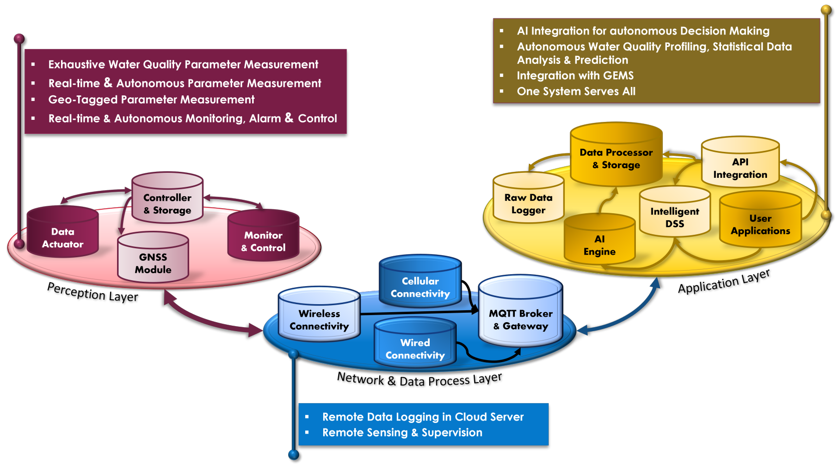

Fig. 1. System Architecture (abstract) mapped with the system requirements

The build-in *SD Card* slot is used for the temporary storage of the sensor and Geo-location data, along with other system specific information as required (e.g., system diagnosis data, error code, and user access detail). This data storage allows to ensure no data loss, and load balancing for the network layer in case of system downtime.

The **GNSS & Cellular Module** is connected to the RaspberryPi, provides the geo-location (Longitude and Latitude) for geo-tagging the sensor data in the most privacy preserving way (refer to Figure 2). This module also facilitates the cellular services, e.g., messaging, calling services of the 2G, 3G and 4G network. For instance, sending a SOS call to the designated authorities in case of any alarming condition detected that needs to be instantly notified. The *SIM7600X 4G HAT module* is used for this system development due to its' reliable location measurement while deployed in the outdoor condition [7].

Finally, the **Monitor & Control** unit supports onsite monitoring and controlling of the system. It should offer real-time visualisation of all necessary parameters, analytic and control through wireless or wired communication with the micor-controller. The *HMI display device* with all necessary components (e.g., flash memory, SD card Slot, GPU, RGB Buffer, RGB Driver, Touch Screen, UART Interface) is used for this module. The custom-built S/W module should be developed and deployed in this device.

Furthermore, the payload need to be calculated to determine the tentative maximum data load that the Perception Layer(s) will generate and transmit over the Network Layer. This load calculation will support the design of the Network Layer, the selection of the H/W and the server configurations. To calculate this load following system configuration is considered, (A) 8 digital sensors are connected to the actuator and each sensor generates 4 bytes (Int type) of data, (B) 10 handheld manual sensors each generating 8 bytes of data, (C) Laboratory based data for 54 parameters that are manually fed using the monitor unit, (D) the GPS location data, and (E) the Control signals. With this configuration, the maximum data load generated by a single Perception Layer is approximately 0.55 Kilobytes as calculated in Table I.

Now, considering the Perception Layer is deployed in the 64 cites and all the layers are transmitting data simultaneously in every second. Then the total payload

TABLE I. MAX DATA LOAD GENERATED BY A SINGLE PERCEPTION LAYER

(A) Sensor Generated Data 8 x 4 (B) =	32 B
(B) Manual input sensor data from sensor module 10 x 8 (B) =	80 B
(C) Laboratory based measurement manual input 54 x 8 (B) =	432 B
(D) GPS Signals =	15 B
(E) Control Signals 3 (modules) X 16 (b) =	8 B
Data for continuous transmission (A+B+C+D+E)=	567 B
=	0.55 KB (approx.)

for the entire system would be:

0.55 (KB / Perception Layer) x 60 sec x 64 (cities) =	2112 KB / Min
=	2.0625 MB / Min

Therefore, the maximum payload for the **WQMS** system (considering 64 perception layers are active simultaneously) would generate 2.0625 MB of data per minute, which need to be handled by the Network Layer.

B. Network and Data Process Layer

This layer establishes a stable and secure network inter-connectivity between the Perception Layer and the Application Layer over the internet. However, due to the drifts in IoT data traffic and unique network requirements among the IoT systems, the design and development of this layer differs significantly [8].

For the WQMS system, the volume of data that need to be exchanged is approximately ∼**2.1 MB/Minute** as calculated in Section IV-A. This data should be transmitted by the Network Layer to the central cloud server hosted in the Application Layer. The vast majority of this data traffic is in the form of periodic transmissions of several lines of text (e.g. Json, XML content) containing sensor measurements, geo-coordinates, toggle switch positions or simple commands. This implies that very low bandwidths are necessary at the access layers of the network. Therefore, the MQTT protocol is adopted due to its' capability to handle a maximum of 256MB of data/Minute (small packet less than 127bytes) which is significantly higher than the load calculated for this system to handle.

However, handing of thousands of connected devices (more than 2000 sensors and associated devices are estimated for the **WQMS** system) poses a design challenge for this layer [2], [9]. This includes, reliable and

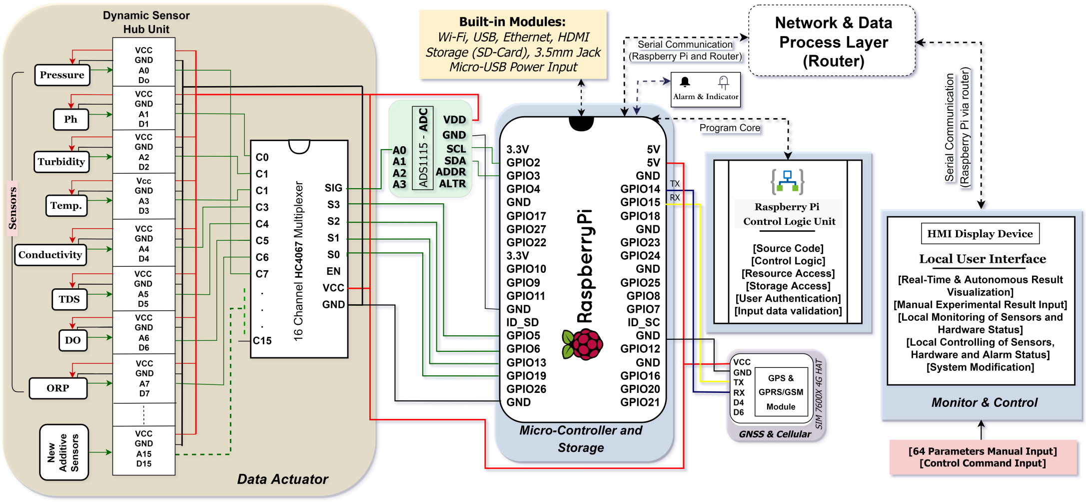

Fig. 2. The Hardware Configuration and Design to conceptualize / implement the Perception Layer

efficient communication with high degree of availability, scalability, alternation and switching with adequate data security [9]. Considering these concerns, the design and implementation of the *Network Layer* for the **WQMS** system is presented in Figure 3.

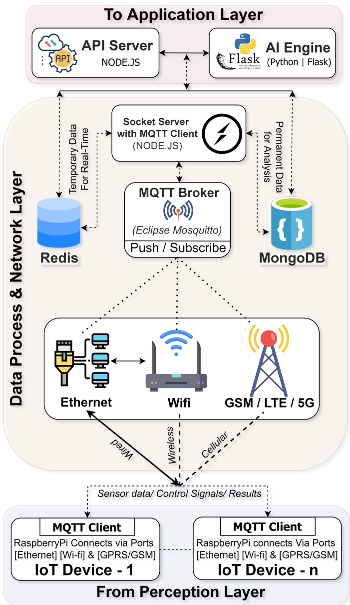

Fig. 3. Conceptualization and Internal Components of Network and Data Process Layer

A hybrid network topology is used to design this layer, within which alternate network connectivity along with communication channel switching is implemented to ensure availability and scalability. As shown in Figure 3, this layer is equipped with three types of network connectivity in parallel, namely, the traditional broadband connection (wired), the GSM/LTE/3G/4G/5G networks (cellular connectivity), and the WiFi Router (wireless connectivity).

From the **Perception Layer**, the integrated Ethernet ports and the WiFi module of the Raspberry Pi micro-controller, and the Cellular module with built-in GPS, GPRS/GSM/LTE communication protocols are used to establish connection with the **Network Layer** (as shown in the lower part of Figure 3).

Towards the **Application Layer**, a Socket Server with MQTT Client is implemented in the Network Layer to establish communication. This server is connected with two data servers implemented in the Application layer, namely, the Radis Server and the MongoDB. Having these two servers in place, facilitates both real-time synchronous and asynchronous data logging that are received from the designated sources. These communication arrangements are shown in the upper part of Figure 3. Within the *Application Layer*, the API server and the AI Engine get access to the data stored in these two servers (detail is discussed in Section IV-C).

To establish the communication among these hardware systems, a communication protocol need to be implemented within the Network Layer. Therefore, the MQTT server with the MQTT Broker is implemented to control the data transmission. The task of the MQTT broker is to receive and filter the data send by the Publisher and transmit it to the Subscriber. For this implementation, the Eclipse Mosquitto message broker is used. In this arrangement, the devices that want to send data, get registered as the Publishers, and the one that wants to receive it, get registered as the Subscriber with the MQTT Broker. For instance, while sending data from the Perception Layer to the Application Layer, the micro-controller first get registered as a Publisher with the MQTT broker, and then transmit the data.

C. Application Layer

Within the **WQMS** system, the Application Layer should perform the followings: (a) logging of geo-tagged raw sensor data that are periodically transmitted by the *Perception Layers* deployed over several cites, (b) pre-processing of the sensor data and storage in the databases, (c) perform specialized data processing, water quality assessment and statistical evaluations, (d) management for the end-user services, (e) implement and execute AI powered autonomous data processing, assessment, visualization, and decision making, (f) API integration for data access, third-party sharing and exchange, and (g) run user applications for remote monitoring, supervision, controlling and research. The elaborated design for this layer is presented in Figure 4.

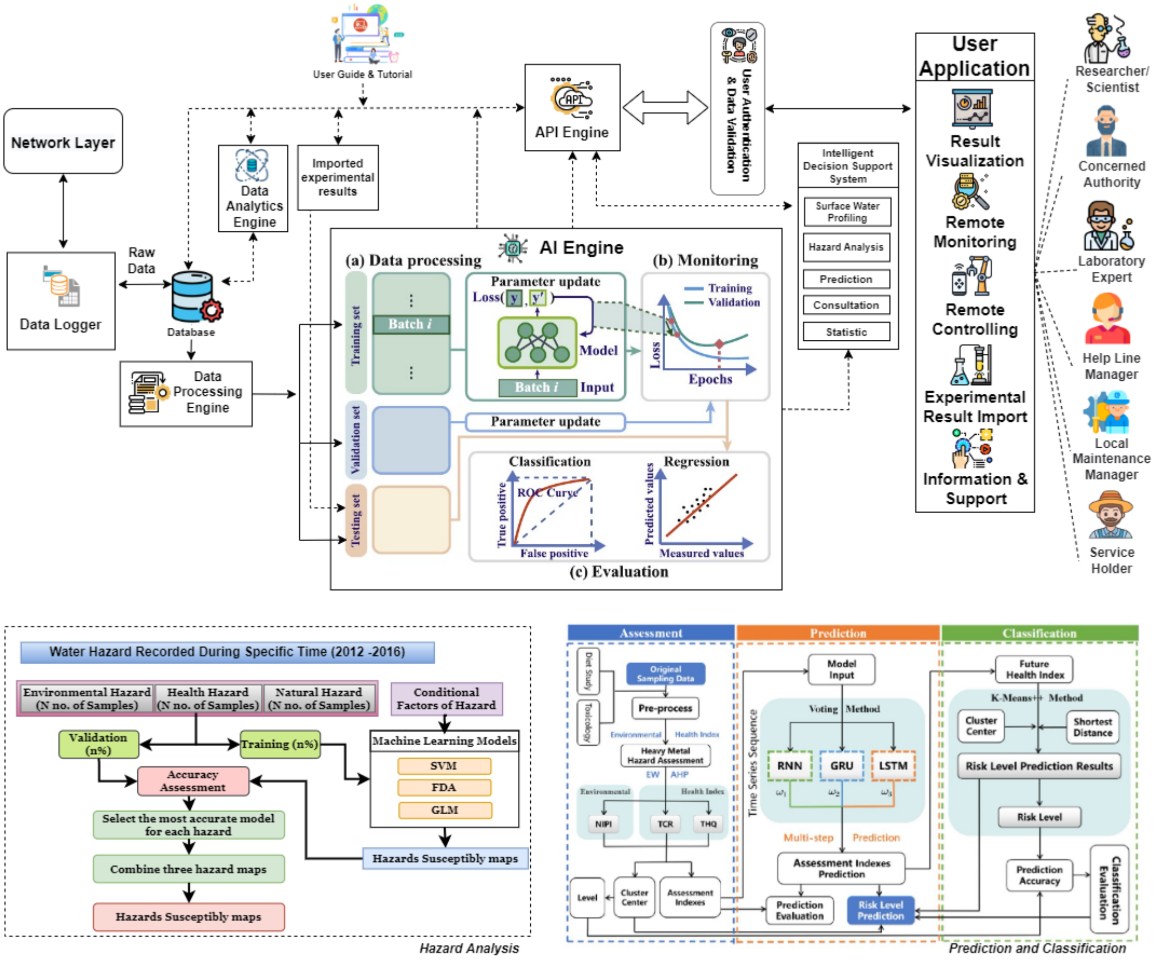

Fig. 4. Internal Components of Application Layer

In order to develop this layer, a cloud centric architecture underpinned by the fog computing paradigm is adopted, so that the data processing can be done in a large centralized fashion through dynamic resource allocation [5], [10]. Additionally, it has a very flexible interface to deploy all the necessary tools for storage, software, development, visualization, data mining, machine learning, and AI integration. Commensurate with Figure 4, the essential modules for this layer are: the fog computing layer, the databases, data analytic engine, the AI engine, the API server, the intelligent decision support system, and the user applications for targeted user groups along with supporting guides and tutorials.

The use of **Fog computing** ensures pre-processing of the sensor data received through the network layer on the edge of the network, and thus, increasing data processing efficiency, and reduce the load of the main cloud servers [11], [12]. It consists of a Socket Server with MQTT client and the Data Processing engine. The socket client is implemented in *Node.js* that establishes the connection with the Network Layer for sending commands or receiving the data. The Data Process Engine is used to parse, join, filter, and curation of the data into well structured tabular format and store it in the persistent database server (implemented in MongoDB). Other modules in this layer, e.g., the AI engine can readily fetch the data from this server to train the ML models. This arrangement is shown in Figure 4.

The **Database module** consists of a Redis database and a MongoDB database. The Radis database store data in the RAM and is volatile, therefore, only used for real-time and fast access of the data. However, for the persistent storage and long-term data use, the MongoDB database is used. The **Data Analytic Engine** is a multipurpose server module that performs three tasks: (a) perform *Descriptive Analytic* for summarising and report preparation from water quality assessment, (b) perform *Predictive Analytic* on the water quality data to predict future progression and patterns, and (c) perform *Prescriptive Analytic* to provide consultation and decision based on specific observations. These analysis results are often logged into the MongoDB database for use by the other modules. For instance, the AI engine can extract the data for training the ML models. Also, the API Engine can execute appropriate APIs to stage the data for retrieval. The staging may include use case evaluation, data identification, filtering, extraction, aggregation, analysis and visualization.

The integration of the **AI Engine** is due to assimilate intelligence for doing the tasks without human intervention. Knowing the fact that the WQMS system will produce huge amount of data which requires constant analysis for classification and real-time decision making, predictive analysis, trend and pattern identification, and maintenance of the IoT devices, this integration of AI is an indisputable engineering design preference. There are several ML (Machine Learning) algorithms that can be applied in WQMS system for the model development. ML algorithms can be leveraged to train AI models with the data for (a) classification of the water quality profiling based on geo-locations and other classification parameters, (b) predict future behaviour, pattern, events and anomalies based on the past behavioural trends, (c) deliver insights otherwise hidden in data for rapid, real-time and automated responses, and improved decision making. The **AI Engine** should also support the implementation and deployment of both *Supervised learning* (e.g., linear and non-linear Regression, classification models) and *Unsupervised learning* (e.g., hierarchical clustering, K-means clustering, Principal Component Analysis) based ML models. Figure 4 presents a manifestation of a classification model for Hazard and Risk analysis for the fisheries based on hazard indexes. For the development of the ML models, the python based ML frameworks, libraries and popular IDEs should be made available in the cloud server. With authorisation, developers and researchers can use these facilities for model development and use.

The **Intelligent Decision Support System (IDSS)** module is responsible for implementing the business logic required for taking intelligent and smart decision on behalf of the end users. As can be seen from Figure 4, this module can get access to the analytic results (from the data analytic engine) by invoking appropriate APIs through the API server. This includes but not limited to the surface water profiling, hazard analysis, prediction, and statistical analysis results. These data are then fed into the appropriate business logic to derive decisive decisions, observations and reporting.

The **API Engine/Server** is the middle-ware that exposes the APIs (Application Protocol Interface) for unified, standardized and controlled access of data stored in the MongoDB database. The WQMS system requires that all access to the data must be made through API calls, and the API engine should perform this task.

The **User Applications** are the custom made end-user S/W products in the form of mobile, desktop or web-based applications. These applications must acquire session based user authentication token issued by the *User Authentication & Data Validation* module, and then invoke the registered APIs from the API engine for the data access. The service domain of these applications are multi-fold. As listed in Figure 4, applications can serve real-time, interactive visualisation of the data and analysis report, or can offer remote monitoring and controlling of the system (e.g., adjusting the control parameters and controlling the status of the sensors and actuators), or can run scripted experiments on the data to meet diverse needs of the targeted research communities.

Alongside the applications, the user groups can also be segmented and registered for unified and focused access to the data, and the system resources. Figure 4 presents a non-exhaustive list of user groups that includes, researchers, concerned authorities, system operators, among others. Finally, the **User Guide & Tutorial** module offers the online technical and system operation guide to assist the targeted user groups to effectively use the system.

V. Conclusions

This study demonstrates an IoT intensive and AI integrated system design and development for remote sensing and management of the surface water quality. This system consists of a novel plug-and-play physical layer for the sensors and GNSS module to acquire geo-referenced sensor data. The system also has a cloud centric fog computing application layer that supports remote sensing, data logging, monitoring and control, with AI driven statistical and predictive data analysis for critical reasoning and decision making. This system also exposes APIs to be used in an authenticated way by the third parties (e.g., research organisations, universities, govt. authorities,) to access the data and services. Overall, this system can replace the current semi-automated, human intensive and time consuming process of water quality management with a technology intensive intelligent system to do the same. Such a system will facilitate the authorities to automatically assess and predict the contamination, and derive factual observations to plan for sustainable use of the water resources, and associated disaster management.

References

[1] MMM Syeed, Md. S. Hossain, Md R. Karim, Md. F. Uddin, M. Hasan and R.H. Khan, Surface water quality profiling using the water quality index, pollution index and statistical methods: A critical review, Environmental and Sustainability Indicators, 2023. https://doi.org/10.1016/j.indic.2023.100247

[2] Md. A. Islam, R.H. Khan and MMM Syeed. A Smart and Integrated Surface Water Monitor System Architecture: Bangladesh Perspective. In Proceedings of ICCA 2020. ACM, Article 74, 1–6. https://doi.org/10.1145/3377049.3377103

[3] T.G. Nguyen and T.H.N. Huynh, Assessment of surface water quality and monitoring in southern Vietnam using multicriteria statistical approaches. Sustain Environ Res 32, 20 (2022). https://doi.org/10.1186/s42834-022-00133-y

[4] R. El Behairy, A. El Baroudy, M. Ibrahim and M. Shokr, Assessment and mapping of surface water qualtiy index for irrigation purpose: Case study northwest of Nile Delta, Egypt. Menoufia Journal of Soil Science, 2021, 6(5), 163-182. doi: 10.21608/mjss.2021.182346

[5] A. Adnan, M. Hartikainen, U. Hassan, K. Heljanko, J. Lilius, T. Mikkonen, I. Porres, MMM Syeed and S. Tarkoma. Introduction to cloud computing technologies. Developing Cloud Software: Algorithms, Applications, and Tools 60 (2013): 1-41.

[6] K.A. Piirainen, R.A. Gonzalez, Seeking constructive synergy: design science and the constructive research approach. In International conference on design science research in information systems. Springer, 59–72, 2013.

[7] L. Chen et al., Robustness, Security and Privacy in Location-Based Services for Future IoT: A Survey, IEEE Access, vol. 5, pp. 8956-8977, 2017, https://doi.org/10.1109/ACCESS.2017.2695525.

[8] M. Mrabet, s. Belguith, A. Alhomoud,and A. Jemai, A Survey of IoT Security Based on a Layered Architecture of Sensing and Data Analysis. Sensors, 2020, 20(13), p.3625. Available at: https://doi.org/doi:10.3390/s20133625.

[9] S. Samuel, A review of connectivity challenges in IoT-smart home. 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC), 2016, https://doi.org/10.1109/ICBDSC.2016.7460395.

[10] R.E. Guinness, H. Kuusniemi, J. Vallet,T. Sarjakoski, J. Oksanen, J., M. Islam, MMM Syeed, MyGeoTrust: A Platform for Trusted Crowdsourced Geospatial Data, Proceedings of the 28th International Technical Meeting of the Satellite Division of The Institute of Navigation, 2015, pp. 2455-2469.

[11] I. Stojmenovic and S. Wen, The fog computing paradigm: scenarios and security issues, in Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS '14), pp. 1–8, IEEE, Warsaw, Poland, September 2014.

[12] M. Aazam and E.N. Huh, Fog computing and smart gateway based communication for cloud of things, 2nd IEEE International Conference on Future Internet of Things and Cloud (FiCloud '14), pp. 464–470, Barcelona, Spain, August 2014.