Real -time water level data and flood warnings

# Introduction:

## **Abstract**

Today, a large portion of the human population around the globe has no access to freshwater for drinking, cooking, and other domestic applications. Water resources in numerous countries are becoming scarce due to over urbanization, rapid industrial growth, and current global warming. Water is often stored in the aboveground or underground tanks. In developing countries, these tanks are maintained manually, and in some cases, water is wasted due to human negligence. In addition, water could also leak out from tanks and supply pipes due to the decayed infrastructure. To address these issues, researchers worldwide turned to the Internet-of-Things (IoT) technology to efficiently monitor water levels, detect leakage, and auto refill tanks whenever needed.

## **1. Introduction**

Water is necessary for life on our planet. Seventy-one percent of the earth’s surface is covered with water. No doubt, this quantity is much smaller compared with the total earth volume [[**1**](https://www.mdpi.com/2073-4441/14/3/309#B1-water-14-00309),[**2**](https://www.mdpi.com/2073-4441/14/3/309#B2-water-14-00309),[**3**](https://www.mdpi.com/2073-4441/14/3/309#B3-water-14-00309),[**4**](https://www.mdpi.com/2073-4441/14/3/309#B4-water-14-00309),[**5**](https://www.mdpi.com/2073-4441/14/3/309#B5-water-14-00309),[**6**](https://www.mdpi.com/2073-4441/14/3/309#B6-water-14-00309),[**7**](https://www.mdpi.com/2073-4441/14/3/309#B7-water-14-00309),[**8**](https://www.mdpi.com/2073-4441/14/3/309#B8-water-14-00309),[**9**](https://www.mdpi.com/2073-4441/14/3/309#B9-water-14-00309),[**10**](https://www.mdpi.com/2073-4441/14/3/309#B10-water-14-00309),[**11**](https://www.mdpi.com/2073-4441/14/3/309#B11-water-14-00309),[**12**](https://www.mdpi.com/2073-4441/14/3/309#B12-water-14-00309),[**13**](https://www.mdpi.com/2073-4441/14/3/309#B13-water-14-00309),[**14**](https://www.mdpi.com/2073-4441/14/3/309#B14-water-14-00309)]. The oceans contain around 97% of the total water on earth [[**15**](https://www.mdpi.com/2073-4441/14/3/309#B15-water-14-00309),[**16**](https://www.mdpi.com/2073-4441/14/3/309#B16-water-14-00309),[**17**](https://www.mdpi.com/2073-4441/14/3/309#B17-water-14-00309),[**18**](https://www.mdpi.com/2073-4441/14/3/309#B18-water-14-00309),[**19**](https://www.mdpi.com/2073-4441/14/3/309#B19-water-14-00309),[**20**](https://www.mdpi.com/2073-4441/14/3/309#B20-water-14-00309),[**21**](https://www.mdpi.com/2073-4441/14/3/309#B21-water-14-00309),[**22**](https://www.mdpi.com/2073-4441/14/3/309#B22-water-14-00309),[**23**](https://www.mdpi.com/2073-4441/14/3/309#B23-water-14-00309),[**24**](https://www.mdpi.com/2073-4441/14/3/309#B24-water-14-00309),[**25**](https://www.mdpi.com/2073-4441/14/3/309#B25-water-14-00309),[**26**](https://www.mdpi.com/2073-4441/14/3/309#B26-water-14-00309),[**27**](https://www.mdpi.com/2073-4441/14/3/309#B27-water-14-00309),[**28**](https://www.mdpi.com/2073-4441/14/3/309#B28-water-14-00309)]. Unfortunately, ocean water also has very heavy salt content and thus cannot be used directly for many household needs such as drinking, cooking, etc. [[**29**](https://www.mdpi.com/2073-4441/14/3/309#B29-water-14-00309),[**30**](https://www.mdpi.com/2073-4441/14/3/309#B30-water-14-00309)]. The rest of the total water is available as freshwater [[**31**](https://www.mdpi.com/2073-4441/14/3/309#B31-water-14-00309)].

**Table 1.** Contemporary reviews on smart water monitoring.



* A comprehensive survey of related work,
* Reviewing recent technologies and techniques,
* Exploring existing software and hardware platforms for IoT-WST, and
* Highlighting the cyber security threats.

The remaining of this article is structured as follows: [**Section 2**](https://www.mdpi.com/2073-4441/14/3/309#sec2-water-14-00309) explains the background of the water storage tanks monitoring. [**Section 3**](https://www.mdpi.com/2073-4441/14/3/309#sec3-water-14-00309) explains the research method and offers a critical survey of contemporary work on IoT-WST. [**Section 4**](https://www.mdpi.com/2073-4441/14/3/309#sec4-water-14-00309) elaborates on some technologies and techniques commonly used while designing and developing IoT-WST. In [**Section 5**](https://www.mdpi.com/2073-4441/14/3/309#sec5-water-14-00309), the authors offer specific details about the potential challenges, trends, and limitations of IoT-WST. Finally, [**Section 6**](https://www.mdpi.com/2073-4441/14/3/309#sec6-water-14-00309) concludes this study. [**Table 2**](https://www.mdpi.com/2073-4441/14/3/309#table_body_display_water-14-00309-t002) has the list of acronyms utilized in this article.

**Table 2.** Commonly used acronyms.



## 2. Background

This section briefly details some fundamental basics, which may offer convenience to readers of this article.

#### ***2.1. Traditional Monitoring***

Most often, the water tank monitoring is performed manually [[**59**](https://www.mdpi.com/2073-4441/14/3/309#B59-water-14-00309),[**60**](https://www.mdpi.com/2073-4441/14/3/309#B60-water-14-00309),[**63**](https://www.mdpi.com/2073-4441/14/3/309#B63-water-14-00309),[**73**](https://www.mdpi.com/2073-4441/14/3/309#B73-water-14-00309)]. For example, a consumer can refill a tank when it is empty and fix water leakage if any is detected. Though this method has been in use for a long time, it has some serious limitations. For instance, 24/7 h monitoring of tanks in person may not be feasible for individuals at private locations such as homes, schools, universities, organizations, mills, factories, etc. In fact, it is often the case that water overflows from tanks undetected. Moreover, the task of manually checking the water level in tanks if needed is tedious and often not impossible.

#### ***2.2. Off-line Automated Monitoring***

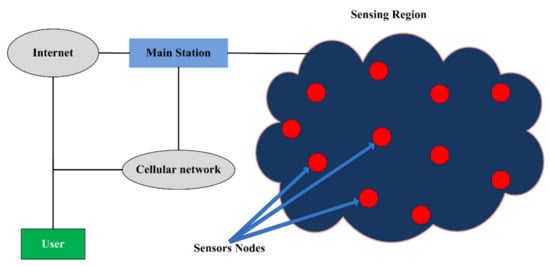
As noted above, manual monitoring of water storage tanks may not be a comfortable experience, especially in the water-scarce regions such as Saudi Arabia [[**31**](https://www.mdpi.com/2073-4441/14/3/309#B31-water-14-00309),[**72**](https://www.mdpi.com/2073-4441/14/3/309#B72-water-14-00309),[**74**](https://www.mdpi.com/2073-4441/14/3/309#B74-water-14-00309)]. Thus, researchers have devised off-line embedded systems to monitor water storage tanks [[**31**](https://www.mdpi.com/2073-4441/14/3/309#B31-water-14-00309),[**32**](https://www.mdpi.com/2073-4441/14/3/309#B32-water-14-00309),[**59**](https://www.mdpi.com/2073-4441/14/3/309#B59-water-14-00309),[**60**](https://www.mdpi.com/2073-4441/14/3/309#B60-water-14-00309)]. In such approaches, researchers deploy a microprocessor-based system for monitoring tanks. A typical off-line tank monitoring system may include (1) sensors, (2) actuators, (3) processor(s), and (4) supportive electronic components. These units are briefed below:

* Sensor: It can detect modifications in its surroundings and transfer collected data to relevant electronic modules (e.g., a microprocessor). Notably, a sensor is always supplemented with other electronic modules (e.g., analog-to-digital converter (ADC)) for proper signal conditioning [[**75**](https://www.mdpi.com/2073-4441/14/3/309#B75-water-14-00309)].
* Actuator: it is a device (e.g., transistor, electromechanical relay, and thyristor), which is capable of causing machines or devices to run.
* Processor: In embedded systems, dedicated microprocessors (also called microcontrollers) are utilized. In general, a microcontroller unit (MCU) is called a true computer on a single chip, which has all necessary peripherals (e.g., memory, timers/counters, digital and analog input/output (I/O) ports, ADC, and digital-to-analog converter (DAC)) on-chip. This unit can easily read sensors data, process, store, and update output devices (e.g., liquid crystal module (LCM)) if required, and transfer data to other devices and machines if needed.
* Supportive electronic components: (e.g., power supply unit, buffers, resistors, and diodes) are always required to power up the target system, integrate sensors with I/O ports, etc.

To summarize, off-line automated monitoring systems are suitable for monitoring water storage tanks, but locally. They may not have the requisite electronic interface or modules to transfer sensor data to remote devices through wired or wireless communication channels. For this reason, their scope is limited to individual usage only.

#### ***2.3. WSN-Based Monitoring***

To extend the capabilities of the off-line monitoring systems, researchers approached towards usage of the WSN technology [[**68**](https://www.mdpi.com/2073-4441/14/3/309#B68-water-14-00309),[**76**](https://www.mdpi.com/2073-4441/14/3/309#B76-water-14-00309),[**77**](https://www.mdpi.com/2073-4441/14/3/309#B77-water-14-00309)]. In a typical WSN ([**Figure 1**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f001)), a sensor node (an MCU-based kit) first reads in the sensors data (e.g., leakage) being installed on-site. After reconditioning and processing, the data is sent to the main station (also called a server) wirelessly using different wireless channels (e.g., Lora WAN, Xbee, Wi-Fi, Bluetooth, nRF24L01, or RF 433) [[**78**](https://www.mdpi.com/2073-4441/14/3/309#B78-water-14-00309)]. On the reception site, the main station performs further analysis to find out hidden patterns and anomalies, if any. Finally, the processed data is stored, results are revised, and, if needed, feedback is initiated to the relevant authorities or end-users via an email, SMS, etc. In this scenario, the main station may also have full control over sensors and actuators being connected to each sensor node.

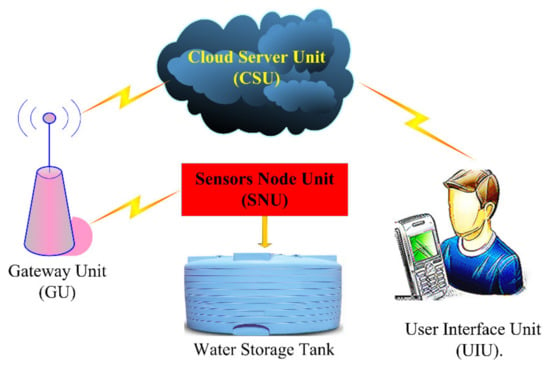


**Figure 1.** A typical block diagram of WSN.

While resolving some of the limitations of off-line monitoring systems, this technology also has some limitations, such as low spatial resolution due to private network infrastructure, compromised security, energy requirements, storage issues, and high maintenance and installation costs.

#### ***2.4. Smart Monitoring***

Due to the spatial limitations of WSN technology, sensor nodes are monitored through a local server. Thus, global access to individual nodes in such networks is impossible. To address this issue, researchers resorted to using IoT technology, i.e., smart monitoring. In such technology, each node can directly send data to an IoT cloud server [[**79**](https://www.mdpi.com/2073-4441/14/3/309#B79-water-14-00309)], or nodes may also forward sensors data to a master node, subsequently transmitting it to the IoT cloud server for further processing, analysis, etc. Moreover, the concerned authority or end-user may also have direct access to each node and may .



**Figure 2.** A typical diagram of IoT-WST.

Briefly, the SNU captures sensors data (e.g., water level and leakage), reconditions and processes it, local displays are updated accordingly and then sent to CSU through a Hotspot (local Wi-Fi), ethernet channel, or GSM/GPRS modem. The GU is responsible for the communication between SNU and CSU. Almost all internet service providers currently facilitate their customers through the hired cloud servers and other facilities whenever needed [[**79**](https://www.mdpi.com/2073-4441/14/3/309#B79-water-14-00309)]. Today, third parties offer cloud servers for many IoT applications [[**79**](https://www.mdpi.com/2073-4441/14/3/309#B79-water-14-00309)], e.g., Ubidots, Blynk, and Adafruit. IoT developers can more comfortably develop an extensive range of IoT products using commercial cloud servers for many useful applications, e.g., health, automobiles, water monitoring, and aquaculture. Some major highlights of this technology are as follows:

1. Reduced Cost: As it uses the existing communication infrastructure of the internet, the overall cost for the system’s development has been reduced, e.g., no personal communication network is generally required.
2. Higher Spatial Resolution: As its backbone is based on the internet, its spatial resolution is ideally infinite. It implies that monitoring water storage tank is possible from any corner of the globe wherever access to the internet is possible.
3. Reduced Computational Cost: In general, a sensor node should be equipped with an ordinary MCU/CPU (Central processing unit) based kit (e.g., NodeMCU [[**85**](https://www.mdpi.com/2073-4441/14/3/309#B85-water-14-00309)], ESP8266 Transceiver [[**85**](https://www.mdpi.com/2073-4441/14/3/309#B85-water-14-00309)] or Arduino Nano 33-IoT [[**86**](https://www.mdpi.com/2073-4441/14/3/309#B86-water-14-00309)]) and any heavy computational load should be shed to IoT cloud servers, e.g., IBM, Adafruit, Blynk, Arduino, and Ubidots IoT platform [[**79**](https://www.mdpi.com/2073-4441/14/3/309#B79-water-14-00309)]. Thus, the use of hi-tech computing devices such as the DE1 SoC FPGA board [[**75**](https://www.mdpi.com/2073-4441/14/3/309#B75-water-14-00309)] and the Raspberry Pi 4 Model-B [[**87**](https://www.mdpi.com/2073-4441/14/3/309#B87-water-14-00309)] could be avoided.

#### ***Smart Storage Tanks: Results and Discussion***

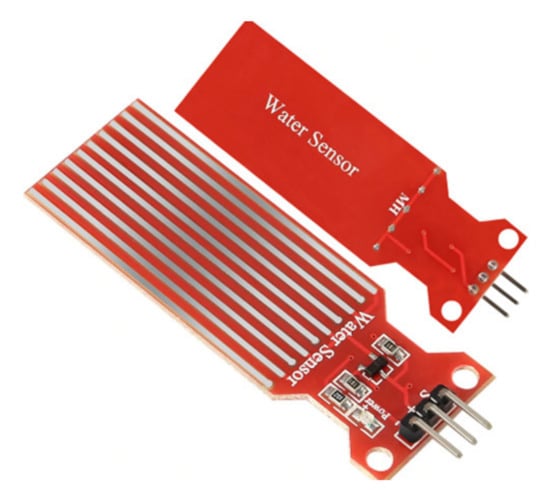
Kumar et al. [[**92**](https://www.mdpi.com/2073-4441/14/3/309#B92-water-14-00309)] published an article focused on developing a microgrid system to control water tanks at the town level. It aimed to reduce water wastage due to tanks overflowing and leakage and the workforce required to monitor water tanks manually. Herein, the tank-mounted unit powered by a solar panel reads water level in the tank via the Arduino Uno kit. This gadget is made of an 8-bit MCU (Atmega328P) surrounded by several peripherals such as the USB interface, ADC, input/output (I/O) pins, timers/counters, SPI (serial peripheral interface) module, etc.

To detect water level, the authors used an ultrasonic sensor (HC-SR04), which is a complete signal conditioning module. Next, they used a GSM/GPRS SIM900A shield plugged in the Arduino kit for transferring data to the ThingSpeak [[**79**](https://www.mdpi.com/2073-4441/14/3/309#B79-water-14-00309)], which is a private IoT cloud server. It can store, analyze, visualize, and find hidden patterns in the acquired sensors data. While some limited resources are offered freely, IoT developers need to pay for the commercial activities if needed.

**Table 3.** Comparison of contemporary IoT-WST.



Nikeeta et al. [[**93**](https://www.mdpi.com/2073-4441/14/3/309#B93-water-14-00309)] proposed an IoT-based water management system to reduce water wastage in residential buildings. The authors considered monitoring water level, leakage control, and auto refilling tank. It is centered around the Raspberry Pi 3 Model B+ kit [[**87**](https://www.mdpi.com/2073-4441/14/3/309#B87-water-14-00309)]. This kit is based on a 64-bit Quad-core processor, @ 1.2 GHz. In addition, it is also abundant in peripherals such as HDMI port, CSI camera port, USB ports, 40 I/O pins, Wi-Fi/Bluetooth/ethernet, 1GB memory, and more. However, it does not offer any on-chip ADC unit; due to this reason, the authors used an eight-channel, 10-bit ADC chip (MCP3008) while acquiring the relevant sensors data.



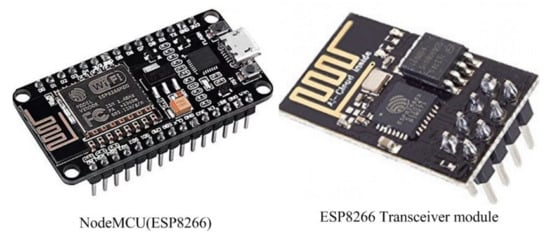
**Figure 4.** Water sensor used in [[**93**](https://www.mdpi.com/2073-4441/14/3/309#B93-water-14-00309)], Courtesy of the Shenzhen Ke Zhi You Technology Co., Ltd., China.

In addition, this article is composed at the abstract level, where some important technical details are also missing. It has the following limitations. Firstly, Raspberry Pi 3 Model B+ 260 (price 260 USD) is costly compared with other IoT kits such as Arduino Nano 33 IoT (price 14 USD) and NodeMCU ESP3266 (prices 24 USD); prices were taken from the Google chrome, dated: 15 September 2021. Secondly, it is clocked @1.2GHz, which is far beyond the requirements of this application. Thirdly, the water sensor ([**Figure 4**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f004)) used here is unreliable; its sensing part is exposed to water, due to which its metallic layers most often get rusty and erode. With this sensor, the resolution of water level is not as satisfactory as it could be obtained using an ultrasonic sensor.

Lade et al. [[**94**](https://www.mdpi.com/2073-4441/14/3/309#B94-water-14-00309)] proposed an IoT-based water management system targeting water regulation in offices and buildings. In this study, both water level and tank auto refilling are monitored. This design is based on Arduino Uno, which works as follows: (1) Arduino Uno measures the water level in the target tank using an ultrasonic sensor; (2) after local processing, data is sent to an ESP8266 transceiver being interfaced with Arduino Uno. The ESP8266 is an SoC (System-on-Chip) having an embedded TCP/IP protocol stack necessary for an MCU while accessing Wi-Fi. In addition, it has four general-purpose I/O pins as well. It sends data to the webserver via a Wi-Fi router.

The cloud server compares this data with the preset tank’s levels. If the water level is found below a lower threshold, the motor pump is turned on via Arduino Uno; otherwise, no action is taken. In case the motor is already on, the data is checked against an upper threshold. If it is found above the preset threshold, the motor is immediately turned off. Though the hardware used here is quite optimized, it has offered nothing to detect water leakage.

Lakshmi et al. [[**95**](https://www.mdpi.com/2073-4441/14/3/309#B95-water-14-00309)] developed an android application for IoT monitoring water tanks. It allows the end-user to monitor the water level in the tank and control water pumps whenever required. This design is also based on Arduino Uno, supplemented with the NodeMCU (ESP8266). Note, the heart of NodeMCU is an ESP8266 transceiver centered around a 32-bit MCU, with numerous peripherals such as a Wi-Fi module, I/O pins, one ADC pin, etc. [**Figure 5**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f005) shows a typical ESP8266 transceiver and NodeMCU (ESP8266) module.



**Figure 5.** NodeMCU (ESP8266) and ESP8266 Transceiver [[**85**](https://www.mdpi.com/2073-4441/14/3/309#B85-water-14-00309)], Courtesy of the Espressif Systems, China.

In this system, Arduino Uno first measures water level through an ultrasonic sensor. After local processing, data is shifted to NodeMCU for uploading to the Blynk IoT platform [[**79**](https://www.mdpi.com/2073-4441/14/3/309#B79-water-14-00309)] through a Wi-Fi router. The authors developed an android application (Blynk App) for end-users’ smartphones to monitor water levels and control water pumps. While using Blynk App on smartphones, end-users can easily monitor water levels and turn the pump on and off whenever required. This system has a few limitations. Firstly, it has no provision to detect water leakage. Another drawback is its hardware redundancy. For example, Arduino Uno is an 8-bit MCU, and NodeMCU is based on 32-bit MCU; thus, one of these could be skipped without any loss of generality.

In [[**96**](https://www.mdpi.com/2073-4441/14/3/309#B96-water-14-00309)], Parimala et al. proposed a solution to monitor water storage facilities based on IoT principles. In this study, only the water levels in tanks are monitored. Arduino Uno supplemented with GSM/GPRS SIM900A shield is the central part of this system. Uno first measures the level of water in the storage container (e.g., Tank) through an ultrasonic sensor via an ADC pin. Next, it processes this data locally and updates a local LCM. In case of any irregularity, it can also turn on the local buzzer. To update data on the web portal, it uses GSM/GPRS SIM900A. End-users can also inquire about water levels through a smartphone. This system was developed in Arduino IDE, and for the webpage, the designing authors used the unified modeling language (UML). It was also validated on the water in the bottle, tank, and pool. Though its architecture and method are well-explained, it is not the best solution for smart monitoring of water tanks. Authors need to incorporate modules for leakage detection and tank auto refill.

Durga et al. [[**97**](https://www.mdpi.com/2073-4441/14/3/309#B97-water-14-00309)] published an article on automatic tank refilling using the IoT addressing the issues such as turning on the water pump when the main water supply and power (electricity) are both available. In addition, they also focused on controlling a situation in which tank filling is in progress when the main supply of water stops or electricity is cut off.



**Figure 6.** Magnetic float sensor [[**100**](https://www.mdpi.com/2073-4441/14/3/309#B100-water-14-00309)], Courtesy of the Dongguan Fuen Electronics Co., Ltd., China.

While monitoring water levels in the tank and controlling the water pump, the authors developed a mobile application using the MIT App inventor [[**109**](https://www.mdpi.com/2073-4441/14/3/309#B109-water-14-00309)]. This tool is essentially a web application, an IDE initially developed by Google and now maintained by the Massachusetts Institute of Technology (MIT). In brief, this tool allows developing applications for smartphones using the web browser and a connected phone or emulator. The main limitations of this scheme involve not using the proper cloud services to analyze sensors’ data and not monitoring the water leakage.

Gupta et al. [[**101**](https://www.mdpi.com/2073-4441/14/3/309#B101-water-14-00309)] proposed an IoT-based system for monitoring and controlling water consumption in residential buildings, organizations, and corporations, where the target area is divided into different blocks (e.g., A, B, C, …, and Z). In each block, tanks are labeled according to building numbers, e.g., Building1, Building2. For each tank, the authors used the NodeMCU (ESP8266) and an ultrasonic sensor to monitor the water level and share data with the server. Every node is equipped with NodeMCU and local Wi-Fi.

To record, analyze, and visualize data, the authors developed numerous databases in the SQL (Structured query language) server. Water in each tank/building is controlled by a motor installed in the maintenance block. Each motor is actuated through an electromechanical relay. If the water in the tank falls below a predefined threshold, the server sends a command to the maintenance block to turn on the corresponding motor. The motor switches off when the respective tank is full. Though this is a good scheme, it has some limitations; the system does not detect water leakage from the tanks. In addition, the authors did not supply specific technical details, raising the question of how the system could be reimplemented if required.

Dissanayaka and Wickramaarachchi [[**102**](https://www.mdpi.com/2073-4441/14/3/309#B102-water-14-00309)] proposed NodeMCU (ESP8266) based system for monitoring water tanks. While an ultrasonic sensor monitors the water level, the tank is auto refilled through NodeMCU interfaced with a motor through a relay module. For sending data to the cloud server, the authors utilized the built-in Wi-Fi function of NodeMCU and a local Wi-Fi router. The water level data, motor on and off status, and the volume of water in the tank are visualized in the Fusion-Chart package, which is accessed through the Firebase real-time databases and tools such as the CSS, HTML (Hypertext Markup Language), and JavaScript. The authors utilized the following Equation (1) to compute the water volume in a circular tank:

𝑉𝑊𝑎𝑡𝑒𝑟=(𝜋𝑟2)𝐿𝑊𝑎𝑡𝑒𝑟,������=��2������,

(1)

with 𝐿𝑊𝑎𝑡𝑒𝑟=ℎ𝑇𝑎𝑛𝑘−𝐿𝐸𝑚𝑝𝑡𝑦������=ℎ����−������. Here, 𝑉𝑊𝑎𝑡𝑒𝑟������, (𝜋𝑟2)��2, 𝑟�, 𝐿𝑊𝑎𝑡𝑒𝑟������, ℎ𝑇𝑎𝑛𝑘ℎ����, and 𝐿𝐸𝑚𝑝𝑡𝑦������ represent the volume of water present in the tank, tank cross-sectional area, tank radius, current water level, tank height, and the total empty portion of the tank, respectively. The same Equation could also be used for the rectangular or other types of tanks but with slight modifications according to the tanks’ structure. End-users can also monitor water tanks via the webpage, which is accessible through smartphones having connectivity to internet. This system did not consider the issue of water leakage.

Natividad and Palaoag [[**103**](https://www.mdpi.com/2073-4441/14/3/309#B103-water-14-00309)] proposed an IoT-based model emphasizing providing a low-cost and efficient system to improve water distribution for communities. Its overall model includes two stages: (i) client site and (ii) control room. In brief, authors in this model tried to monitor water level, auto-refill tanks, regulate water pressure in supply pipes to avoid leakage and burst, and give full access to the control room.

As presented in [**Table 3**](https://www.mdpi.com/2073-4441/14/3/309#table_body_display_water-14-00309-t003), the water level in tanks is gauged through the ultrasonic sensors, pressure in the supply pipes via a 5V analog pressure sensors, water in pipes is regulated through solenoid valves, and the tank is refilled via an electric motor actuated by electromechanical relay. On the client side, all sensors are looked after by Arduino Uno supplemented by a GSM SIM800 modem to report data to a control room server. The authors configured Raspberry Pi3 Model B+ as a server in the control room, equipped with an LED (Light Emitting Diode) monitor, USB mouse, keyboards, and internet/GSM capabilities.

First, Arduino Uno reads all sensors via its I/O ports and transfers data to the control room through a GSM modem. In addition, it can also refill the tank if the water level falls below a lower threshold. On the other hand, the server has also been equipped with the Fuzzy logic algorithm to make a better decision, e.g., refilling a tank, regulating water flow in the pipe, and sending alerts to the overseer or end-users. To summarize, this model is well-planned and provides a good solution to control and monitor water distribution in towns, organizations, corporations, etc. However, a significant limitation of this design is the inadequate leakage detection or control method. Authors only regulate water pressure in the supp

e water level in the tank through stranded wires tied to their base terminals. They inserted an electrical stranded wire tied to the positive terminal of a battery deep into the tank. In addition, they also installed four electrical stranded wires at different points in the tank, where each wire is bonded with the base terminal of a transistor. When the water level rises, the wires connected with transistor bases come in touch with positive voltage through the water as conducting media. This act triggers the concerned transistor to turn on. These states are displayed locally through four LEDs.

To visualize and store the data on the cloud, the authors used the Dweet.io IoT platform, with the JSOUP for API. In addition, they developed an android application for the smartphone to view the water level in the tank. Users have no control of the on/off motor switch, nor can they configure the water level. System programming was completed in the Energia IDE. In terms of its limitations, the system has no strategy for monitoring water leakage and can monitor only four discrete water levels in the tank. Most significantly, the electrical stranded wires used to detect water levels were undoubtedly a cost-effective short-term solution but may soon suffer from the biofilm effects and erosion due to the organic and inorganic materials present in water. In addition, the authors directly inserted a reference electrical wire tied to a positive battery terminal which may cause overconsumption of the battery energy.

Charles et al. [[**107**](https://www.mdpi.com/2073-4441/14/3/309#B107-water-14-00309)] proposed an IoT-based system integrated with the LABVIEW software to monitor water levels in tanks for large areas.This system can monitor a number of tanks simultaneously. The authors used an ultrasonic sensor for each tank to detect water level and NodeMCU for transferring sensors data to the Google cloud server [[**49**](https://www.mdpi.com/2073-4441/14/3/309#B49-water-14-00309)] via a local Wi-Fi router. In the Google IoT platform, data is stored and analyzed. In addition, each tank is equipped with a water pump and a solenoid valve. While controlling water level and refilling tanks, authors developed a separate station centered around the National instrument data acquisition card (NI DAQ (USB2009)). This card is interfaced with a desktop PC through the USB port. For this station, programming was performed in LABVIEW, a graphical programming media. The NI DAQ has many analog and digital I/O pins to which each tank’s solenoid valve and water pump are connected.

First, NodeMCU reads the water level from the tank through an ultrasonic sensor, reconditions and updates data into the Google IoT platform. Next, the main station fetches data from the Google IoT platform and determines if any valve/motor needs to be served. While using LABVIEW and NI DAQ is a promising idea, the authors also introduced significant redundancy in the overall system hardware. Each tank is equipped with NodeMCU, a perfect choice for many standalone IoT applications because it has numerous futures such as the Tensilica 32-bit RISC CPU Xtensa LX106, ADC, DAC, digital I/O pins, SPI, I2C, flash memory, SRAM, USB interface, and the PCB antenna. Therefore, there is no need to control the valve and water pump using the NI DAQ card. Besides, they did not monitor water leakage and could not provide access to end-users, except via the main station only.

Asif et al. [[**108**](https://www.mdpi.com/2073-4441/14/3/309#B108-water-14-00309)] proposed an IoT solution to monitor the household network of water. In this article, the authors monitored three parameters: the water level, the tank auto-refill, and water leakage. In addition, they also monitored the ho clean the tank was through a hybrid of LED and a light-dependent resistor (LDR) installed inside the tank. For leakage detection, they installed water flow sensors supplemented with NodeMCU at each branch of the water distribution system. In addition, they used an ultrasonic sensor to monitor water levels in the tank. Data is recorded into the ThingSpeak IoT platform using local Wi-Fi, and end-users can access it using smartphones. Water consumption and leakage detection are supervised using sensors data fusion and machine learning (ML) schemes. For example, data from water-flow sensors installed near the water tap and ultrasonic sensors could be used to predict water consumption.

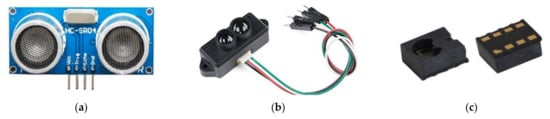
While developing an ML scheme for leakage detection, authors first collect data from the sensors. Then, assuming no leakage occurred for some definite time, upload the data to the ThingSpeak IoT platform, and train the model to infer the decision-marking parameters. In this case, the authors used an implicit relationship between the water height in the tank and water flow in pipes. If leakage is found, the system informs the end-users via smartphones. To monitor water usage, data from sensors is collected and uploaded into the ThingSpeak. Full and empty tank status is also reported to end-users via mobile application. In addition, data from the LED/LDR circuit is used to predict the tank dirtiness; status is reported to end-users.

## 4. Recent Technologies and Techniques

This section presents several well-known technologies and techniques commonly used in the IoT–WST.

#### ***4.1. Water Level Monitoring***

Monitoring the water level in the tank is considered an essential parameter in smart monitoring of water storage tanks. As both cemented and portable tanks exist in different capacities and heights, precise water level measurement may be challenging due to the sensors’ range limitations. As per contemporary literature [[**65**](https://www.mdpi.com/2073-4441/14/3/309#B65-water-14-00309),[**69**](https://www.mdpi.com/2073-4441/14/3/309#B69-water-14-00309),[**101**](https://www.mdpi.com/2073-4441/14/3/309#B101-water-14-00309),[**102**](https://www.mdpi.com/2073-4441/14/3/309#B102-water-14-00309),[**103**](https://www.mdpi.com/2073-4441/14/3/309#B103-water-14-00309),[**107**](https://www.mdpi.com/2073-4441/14/3/309#B107-water-14-00309),[**109**](https://www.mdpi.com/2073-4441/14/3/309#B109-water-14-00309)], some commonly used devices for level monitoring are a water sensor ([**Figure 4**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f004)), magnetic float sensor ([**Figure 6**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f006)), ultrasonic sensor ([**Figure 7**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f007)a), and light detection and range (LIDAR) sensors ([**Figure 7**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f007)b).



**Figure 7.** (**a**–**c**) typical models of an Ultrasonic sensor (e.g., HC-SR04), TF-mini-LIDAR (e.g., SJ-GU-TFmini-01), and Vertical-Cavity Surface-Emitting Laser (e.g., VCNL36826S), [[**86**](https://www.mdpi.com/2073-4441/14/3/309#B86-water-14-00309)]; Courtesy of the Holykell Technology Company Limited, China; Shenzhen FEETECH RC Model Co., Ltd., China; and the Frankfurt Laser Company—Friedrichsdorf, Germany.

While monitoring the water level in tanks, developers also involved discrete type sensors (e.g., water sensor, [**Figure 4**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f004)) that are generally installed at different points inside a tank. As mentioned earlier, in this case, the spatial resolution of the water level is not granular. More sensors need to be installed at different points to enhance the resolution, which is likely an expensive task and may increase the overall burden of maintenance and product cost.

To resolve this issue, researchers involved the distance (proximity) sensors. Such sensors use different technologies such as infrared (IR) triangulation, laser, ultrasonic, LED-TOF (light-emitting diode-time-of-flight), and others. Notably, the choice of the sensor strongly depends on the model of the target application. These sensors offer a range of different properties, e.g., resolution, frequency, field-of-view (FOV), transmission–reception durations, installation, and costs. To detect objects in proximity, these sensors first send waves (e.g., laser, infrared, or sound waves) and then wait for the reception of waves bounced back by the target objects. On reception, the distance of the target objects is gauged based either on the intensity or time taken by the waves to hit objects and come back to the source.

[**Figure 7**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f007)a–c show typical models of the ultrasonic, TF-mini-LIDAR, and a fully integrated proximity sensor (VCNL36826S), respectively. Though each of these sensors can provide acceptable spatial resolution while monitoring water level, the range of LIDAR is relatively higher, as highlighted in [**Table 4**](https://www.mdpi.com/2073-4441/14/3/309#table_body_display_water-14-00309-t004). As LIDARs are costly, these devices are only recommended when the height of a water storage container is more than 4 m. To summarize, proximity/distance sensors are reliable tools for various applications requiring fast and accurate measurement, positioning, or discovery of solid or liquid matter.

**Table 4.** Typical ultrasonic, LIDAR, and VCSEL proximity sensors.



#### ***4.2. Water Leakage Monitoring***

In water-scarce countries, water is often stored in the overhead or underground tanks, filled through the water supply lines from a local municipality, tube wells, or bores. It is observed that water is often wasted from the tanks, supply lines, or tubes due to the aging infrastructure, misuse, or some other technical issues. In addition, water is also wasted through lack of attention and care, causing overflowing of water tanks. The following briefly explains some commonly used water leakage detection techniques:

* Portable Sensors: In numerous studies [[**59**](https://www.mdpi.com/2073-4441/14/3/309#B59-water-14-00309),[**60**](https://www.mdpi.com/2073-4441/14/3/309#B60-water-14-00309),[**110**](https://www.mdpi.com/2073-4441/14/3/309#B110-water-14-00309)], water sensors were used ([**Figure 4**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f004)) to detect water leakage. Generally, one sensor is sufficient for each potential point where water leakage is strongly expected to occur. As mentioned earlier, this type of sensor has two sets of parallel naked metallic layers, where one set is connected with a positive voltage terminal and the other one tied to the negative voltage terminal of the power supply. When its sensory part meets water, this event changes the analog output of this sensor.

Moreover, IoT developers [[**59**](https://www.mdpi.com/2073-4441/14/3/309#B59-water-14-00309),[**60**](https://www.mdpi.com/2073-4441/14/3/309#B60-water-14-00309),[**110**](https://www.mdpi.com/2073-4441/14/3/309#B110-water-14-00309)] also utilized water moisture sensors ([**Figure 8**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f008)) while detecting water leakage. As shown in [**Figure 8**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f008), these are of two types. The one shown in [**Figure 8**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f008)a is not sufficiently reliable because it may rust when its naked metallic layers are exposed to water. The second type ([**Figure 8**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f008)b) is more reliable as its sensing unit has no direct exposure to water.



**Figure 8.** (**a**,**b**) a typical YL-69 water sensor and a capacitive soil moisture sensor, respectively [[**86**](https://www.mdpi.com/2073-4441/14/3/309#B86-water-14-00309)], Courtesy of the Shenzhen Haiwang Sensor Co., Ltd., China.

* **Water Flow Sensors**: [**Figure 9**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f009) shows typical hall-effect-based flow sensors, which are used in many applications such as measuring the flow of water and oil, DIY coffee machines, water vending machines, etc. These sensors come in different sizes and ratings, per the system’s requirements. The main components of a flow sensor include the hall-effect sensor, turbine wheel (also called rotor), and magnet. When water flows through the valve, it rotates the turbine that produces the magnetic field. This change in the magnetic field is sensed by the hall-effect sensor, producing square pulses.

To detect water leakage, developers use a set of two flow sensors that are installed at both starting and endpoints of the target pipe. To monitor these sensors, developers use IoT devices (e.g., BBC Micro-Bit, ESP32, NodeMCU, ESP8266, and Arduino Nano 33 IoT). These devices measure the quantity (liter per hour or cubic meter) of water flowing through flow sensors at both points and then communicate data to each other or a master unit using GSM module, Wi-Fi, Xbee, Bluetooth, LoraWAN, nRF24L01, RF334, or ethernet [[**64**](https://www.mdpi.com/2073-4441/14/3/309#B64-water-14-00309),[**100**](https://www.mdpi.com/2073-4441/14/3/309#B100-water-14-00309),[**111**](https://www.mdpi.com/2073-4441/14/3/309#B111-water-14-00309),[**112**](https://www.mdpi.com/2073-4441/14/3/309#B112-water-14-00309)]. If both readings stay approximately the same, it implies no water leakage or theft. Otherwise, either water from the concerned pipe is illegally diverted, or some leakage occurs. To summarize, this technique seems workable because issues related to hidden water supply pipes (e.g., pipes buried in walls) could be monitored more conveniently.

* **Digital Signal Processing:** In addition to the above-mentioned schemes, researchers also devised numerous digital signal processing-based techniques to detect water leakage [[**51**](https://www.mdpi.com/2073-4441/14/3/309#B51-water-14-00309),[**67**](https://www.mdpi.com/2073-4441/14/3/309#B67-water-14-00309),[**68**](https://www.mdpi.com/2073-4441/14/3/309#B68-water-14-00309),[**80**](https://www.mdpi.com/2073-4441/14/3/309#B80-water-14-00309),[**108**](https://www.mdpi.com/2073-4441/14/3/309#B108-water-14-00309)]. Whenever an abrupt leakage occurs in the supply line, this drastic change produces a spike in a digital filter (e.g., Kalman filter). This spike is a sign of water leakage.

As highlighted in [**Section 3**](https://www.mdpi.com/2073-4441/14/3/309#sec3-water-14-00309), leakage from the water tank could be monitored [[**50**](https://www.mdpi.com/2073-4441/14/3/309#B50-water-14-00309)] first by suspending the water supply from the tank to all the sinks for a finite duration. During this period, the water level in the tank is sampled at regular intervals. In this case, any difference in two consecutive samples would be an important indicator of water leakage from concerned tanks. Though this is a workable idea, its drawback lies in blocking water supply to relevant consumers.



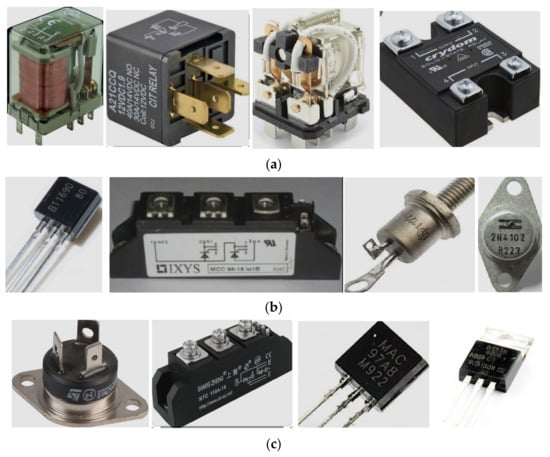
**Figure 9.** Some typical flow sensors [[**75**](https://www.mdpi.com/2073-4441/14/3/309#B75-water-14-00309),[**113**](https://www.mdpi.com/2073-4441/14/3/309#B113-water-14-00309)], Courtesy of the Hunan Mac Sensor Co., Ltd., China; and Ningbo Mingrui Zhongxing Electronics Technology Co., Ltd., China.

#### ***4.3. Tanks Auto Refilling***

Automated refilling of water storage tanks is a feature of IoT–WST. As highlighted in [**Section 4.1**](https://www.mdpi.com/2073-4441/14/3/309#sec4dot1-water-14-00309), the water level in tanks is monitored using portable sensors generally interfaced with computing devices, e.g., NodeMCU, Arduino Uno, and ESP8266 Transceiver.

* Dry-Run: This means the water pump or motor attempts to refill the tank, but water is not available in the source (e.g., supply line from local municipality). This state should be avoided because it may increase the electricity bills.
* Electromechanical Relays: Relays are often utilized to energize water pumps or motors. As shown in [**Figure 10**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f010)a, the layout of each device is printed on its casing, or the device may have a transparent casing through which designers may note its connection. For more details, readers are recommended to consult the technical specifications of target relays on its vendor’s webpage or released documents.
* Thyristors: As relays are electromechanical devices, their performance is defined in terms of switching. In addition, the metallic contact of inexpensive relays (except vacuum type relays) may also become damaged due to the electric arching.

To cope with these issues effectively, IoT developers should use thyristors (e.g., silicon-controlled rectifier (SCR) and TRIAC), [**Figure 10**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f010)b,c. As these devices are based on semiconductor technology, they should function optimally for a long time if appropriately utilized. Notably, controlling water pumps or motors using SCR/TRIAC is complex compared with relays (refer to the proper datasheet of each device before its application).



**Figure 10.** (**a**–**c**) Some typical electromechanical relay, SCRs, and TRIACs, respectively [[**75**](https://www.mdpi.com/2073-4441/14/3/309#B75-water-14-00309),[**113**](https://www.mdpi.com/2073-4441/14/3/309#B113-water-14-00309)], Courtesy of the Zhejiang NCR Industrial Co., Ltd, China; and EASTRONIC THCHNOLOGY CO., LTD., China.

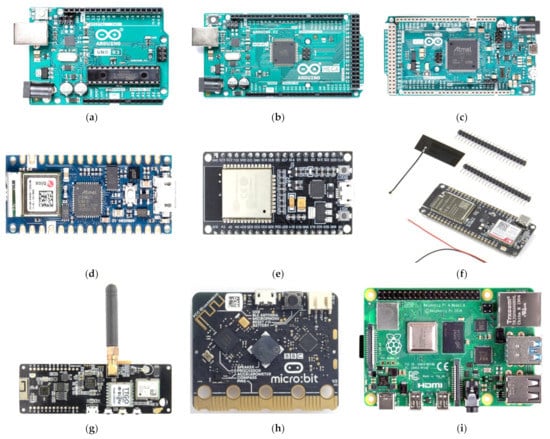
## 5. Challenges and Trends in IoT

Both software and hardware platforms have advanced over the past decades, creating a challenge for IoT developers to select appropriate media while implementing a target project. Extensive experience or a quick but comprehensive review of contemporary solutions and developments in this area may help resolve the difficulty.

Due to the latest advances in the semiconductor industry, our world is now dominated by electronic devices, e.g., portable sensors, tablet PC, cellphones, Wi-Fi, etc. In addition, electronic devices also significantly benefit from miniaturization and improvement in silicon technology, power consumption, portability, and reliability. As developing a hardware system involves cost, IoT developers should select the best choice for the target hardware so that the balance between cost and performance is not a compromise. In addition, software platforms have gone through significant advances, improvements, and modifications over the past decades. For example, new languages and software development platforms have been launched over the past several decades. The following elaborates a few typical challenges IoT developers are facing today.

#### ***5.1. Hardware***

In IoT design, the main task of a digital computing gadget is to read sensors data from its environment, process it locally if required, and then transfer it to the target cloud server via local gateway (Wi-Fi) router, ethernet, or cellular network (i.e., GSP/GPRS module). To accomplish this task, there exist a plethora of commercially available digital development kits, which are based on 8-bit, 16-bit, 32-bit, or 64-bit processors. [**Figure 11**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f011) shows some typically electronic kits often used in IoT designing. [**Table 5**](https://www.mdpi.com/2073-4441/14/3/309#table_body_display_water-14-00309-t005) shows a brief comparison of some expressive features of these kits.



**Figure 11.** (**a**–**i**) Arduino Uno R3 [[**86**](https://www.mdpi.com/2073-4441/14/3/309#B86-water-14-00309)], Arduino Mega 2560 [[**86**](https://www.mdpi.com/2073-4441/14/3/309#B86-water-14-00309)], Arduino Due [[**86**](https://www.mdpi.com/2073-4441/14/3/309#B86-water-14-00309)], Arduino Nano 33 IoT [[**86**](https://www.mdpi.com/2073-4441/14/3/309#B86-water-14-00309)], ESP-WROOM-32S DEV (also called NodeMCU (ESP32S)) [[**85**](https://www.mdpi.com/2073-4441/14/3/309#B85-water-14-00309)], TTGO T-Call ESP32 with SIM800L GPRS module [[**85**](https://www.mdpi.com/2073-4441/14/3/309#B85-water-14-00309)], TTGO LoRa32 SX1276 board [[**85**](https://www.mdpi.com/2073-4441/14/3/309#B85-water-14-00309)], ESP32 TTGO T-Beam V1.1 [[**85**](https://www.mdpi.com/2073-4441/14/3/309#B85-water-14-00309)], BBC Micro: bit V2 [[**114**](https://www.mdpi.com/2073-4441/14/3/309#B114-water-14-00309)], and Raspberry Pi 4 Model B—8 GB [[**87**](https://www.mdpi.com/2073-4441/14/3/309#B87-water-14-00309)], respectively; Courtesy of the Sunhokey Electronics Co., Ltd., China; Espressif Systems, China; and Zhongshan Baijia Dagu Electronic Technology Co., Ltd, China.

**Table 5.** A comparison of contemporary electronics development kits.



A large number of contemporary development kits, e.g., Arduino Uno R3, Arduino Mega2560 [[**86**](https://www.mdpi.com/2073-4441/14/3/309#B86-water-14-00309)], are centered around the 8-bit processor (more technically, these processors are called microcontrollers). These kits are more suitable for the less complex computational designs; they cannot be expected to run any AI or ML algorithms because they work at the lower frequency, e.g., 16MHz (see [**Table 5**](https://www.mdpi.com/2073-4441/14/3/309#table_body_display_water-14-00309-t005)). In addition, most of these kits have no Wi-Fi, Bluetooth, and GSM/GPRS modules onboard or on-chip to shift data directly to the target cloud server if required.

To cope with these limitations, such kits are usually equipped with a suitable module such as the ESP8266 transceiver, GSM/GPRS module, Zbee, Bluetooth, and others. Moreover, such kits offer less memory, prompting developers to use flash memory or SD Cards to store valuable data. Notably, most of these kits are best suited for many embedded applications such as automobiles and home appliances and are abundant in numerous peripherals, including ADC, DAC, timer/counters, touch pins, PWM pins, UART, I2C, and SPI, among others. Though 16-bit processor-based kits are relatively better than 8-bit kits, they are not suitable for modern designs that inherit complex algorithms such as AI and ML.

There is a strong trend of using 32-bit processor-based electronic development kits in the latest IoT design and development. In the context of current IoT requirements, these kits are usually equipped with numerous onboard communication modules such as Wi-Fi, Bluetooth, low energy (BLE), LoraWAN, Zbee, or GSM/GPRS. Using these modules, these kits can send environmental data directly to the cloud server via a local gateway or cellular network.

[**Figure 11**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f011)c shows Arduino Due, which is centered around a 32-bit processor. Compared with Uno and Mega, this kit is extremely rich in peripherals such as digital I/O, PWM, ADC, DAC, UART, SPI, I2C, and memory, among others. In addition, it runs on 84MHz, which implies this device is suitable for the less complex computational AI/ML applications. This device has no module to send data directly to the internet, so it is equipped with an ESP8266 transceiver or a GSM/GPRS module. However, it is less attractive than its siblings (e.g., Arduino Nano 33 IoT ([**Figure 11**](https://www.mdpi.com/2073-4441/14/3/309#fig_body_display_water-14-00309-f011)d), now well-equipped with the needs and requirements of modern IoT designs.

The demand for IoT products increases exponentially due to its vast applications, e.g., smart homes, villages, and cities; medicine, agriculture, water monitoring, wearables, spying devices, vehicle tracking, and children monitoring. Numerous stakeholders are now competing, making their devices smarter, more affordable, reliable, and accessible to all customers. In this respect, the ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD. [[**85**](https://www.mdpi.com/2073-4441/14/3/309#B85-water-14-00309)] has launched many electronic devices and kits highly suitable for AI, ML, and IoT applications. For example, ESP8266 transceiver, NodeMCU (8266), NodeMCU (ESP32), ESP32SCAM, TTGO T-Call ESP32 with SIM800L GPRS Module, TTGO LoRa32 SX1276 Board, and ESP32 TTGO T-Beam V1.1 are just some of the products are based on the ESP devices.

As shown in [**Table 5**](https://www.mdpi.com/2073-4441/14/3/309#table_body_display_water-14-00309-t005), ESP devices are abundant in peripherals (e.g., ADC, DAC, UART, SPI, I2C, Touch pins, Wi-Fi, Bluetooth, BLE, timers, counters, and memory). The majority of the ESP chips have dual processor cores, which can operate at higher clock frequencies such as 80~240 MHz.

Most RESSIF technology-based products are well-suited for modern IoT, AI, and IoT applications. For example, the TTGO T-Call ESP32 with the SIM800L GPRS module has Wi-Fi, Bluetooth, BLE, and an onboard GSM/GPRS to send environmental data to cloud server either via Wi-Fi, Bluetooth, BLE, or cellular network using GSM/GPRS module (enabled with an active SIM). The TTGO LoRa32 SX1276 board can send data to other Lora-enabled devices over

Recently, the BBC Micro:bit has gained enough considerations from hobbyists, developers, and scientists worldwide [[**114**](https://www.mdpi.com/2073-4441/14/3/309#B114-water-14-00309)]. Similar to ESP products, this kit is also based on a 32-bit processor that runs at 64MHz. These kits are abundant in peripherals such as SRAM, Flash, I/Os, ADC, DAC, I2C, and MAG3110 3-axis magnetometer. No doubt, this silicon crab offers strong performance, but its relevant forum and communities seem limited compared with the Arduino products.

In addition to 32-bit devices, there are many 64-bit processor-based electronic development kits, such as Raspberry Pi 4 Model B [[**87**](https://www.mdpi.com/2073-4441/14/3/309#B87-water-14-00309)]. No doubt, such boards are more suitable for the IoT, AI, or ML applications, where more computations and throughput are needed locally. Though Raspberry Pi kits are powerful enough, they are costly and require operating systems in the SD cards.

To conclude this section, the authors highly recommend 32-bit electronic development kits (equipped with Wi-Fi, Bluetooth, BLE, Xbee, Lora, or GSM/GPRS) for the design and development of IoT products.

#### ***5.2. Cloud Servers***

In IoT designing, IoT cloud servers play a crucial role. Briefly, the main task of an IoT cloud server is to facilitate IoT developers in storing, analyzing, and visualizing sensors data. In addition, it may also provide complete control over the functionality of the IoT-enabled devices (e.g., water pumps and vehicles) from any corner of the globe. The following elaborates on this aspect of IoT designing.

* Commercial IoT cloud server: Accompanying the current boom in the IoT-enabled products worldwide is a plethora of contemporary IoT cloud servers [[**79**](https://www.mdpi.com/2073-4441/14/3/309#B79-water-14-00309)], such as ThingSpeak, SensorCloud, Blynk, Arduino cloud IoT, IBM IoT, Adafruit io, and others. In general, each platform can process, analyze, and store data. Moreover, most cloud
* s. Other complications may be the usage of poor cryptographic techniques.
* Intrusion Ignorance: After being compromised, the IoT devices often keep working as usual from the user’s viewpoint. Notably, any power or additional bandwidth usage is generally not identified. It is also observed that most IoT devices do not inherit any alerting or logging functionality to inform end-users of security-related issues. In case IoT devices have these functionalities, the concerned hacker can overwrite or disable them when IoT devices are hacked; the end-user cannot take preventive measures to mitigate the effects of cyberattacks.
* Application Weaknesses: To secure IoT devices, it is important to acknowledge vulnerabilities, if any, in the software in the first place. It is observed that software bugs can trigger malicious activity. Cybercriminals can run their own software on IoT devices and gain illegal access to the sensitive information stored on the concerned devices. Though avoiding software vulnerabilities altogether may not be possible, the developers can adopt best programming practices to escape application vulnerabilities, e.g., performing input validation consistently.
* Vendor’s Security Stance: Whenever software vulnerabilities are detected, it is for the sake of utmost reliability that the concerned vendor finds a proper patch to mitigate their effects. In this concern, the IoT vendors should offer their contact information so that end-users and developers can communicate if any bugs or security loopholes are found. Otherwise, the end-users and developers would not cease using IoT devices in the intended method, resulting in less secure IoT systems. To avoid any catastrophic situation, concerned vendors must cooperate with the end-users and developers, providing frequent updates on the security of the IoT devices and recommendations on how to securely resell or dispose of IoT devices so that the sensitive data is not passed on.
* Deficiency of Reliable Execution Environment: Every IoT device is generally equipped with a dedicated microcontroller, which is a true computer on a single chip capable of running specific software programs. The cyber attacker may install any malicious programs. For instance, they can install a software routine performing a DDoS attack. While limiting the functionality of an IoT device and the software it can run, the potential for abuse of the device is limited. For instance, the concerned IoT device may be confined to connecting only to the vendor’s cloud service. No doubt, this act of restriction may make it unproductive in a DDoS attack because such a device can no longer be able to connect target hosts arbitrarily.

The code that is typically signed with a cryptographic hash should be used to address this issue. As only the concerned vendor has the key to sign software, the IoT device will only run the vendor distributed software. The cybercriminal may no longer run their arbitrary code on the target device.

* **Inadequate Physical Security:** Cyber attackers may open the IoT devices and can attack the hardware if they have access to these devices. For instance, hackers can bypass protective software if they can directly read the contents of memory components. After opening up IoT devices, hackers may read the device debugging contacts and perform more fatal tasks. The physical attack may have more impact if it uncovered the device key shared with all IoT devices of the same model; this act would compromise many IoT devices.
* **User Interaction**

IoT vendors should provide proper documentation and guidelines on deploying their products more securely in IoT-based systems. Users must change the default passwords to help prevent hackers from gaining access to their devices.

## 6. Conclusions

Over the past several decades, water resources have been gradually decreasing, becoming scarce in several places worldwide. To cope with this issue, different nations worldwide are now taking serious measures to mitigate the effects of the water crisis. In this respect, smart monitoring of water resources has gained tremendous attention within the research communities.