International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 17 August 2023 **Revised:** 20 September 2023 **Accepted:** 10 October 2023*

Automated Pet Water Consumption Monitoring System

Sundar S[1*],Anurag Mukherjee[2],Shubham Kumar[3],Mohd. Kabir Haider[4],C.M.Vidhyapathi[5],Karthikeyan B[6],Venugopal P[7],

Satheesh Kumar S[8], Vasavi.C.S[9],and Umashankar Subramaniam[10]

[1,6,7]Associate Professor,

School of Electronics Engineering,

VIT University,Vellore,India.

e-mail :[1]sundar.s@vit.ac.in; [6]bkarthikeyan@vit.ac.in; [7]venugopal.p@vit.ac.in

[2,3,4] School of Electronics Engineering,

VIT University,Vellore,India

e-mail:[2]anurag.mukhherjee2019@vitstudent.ac.in;[3]shubham.kumar2019c@vitstudent.ac.in;[4]mohdkabir.haider2019@vitstudent.ac.in

[5,8]Assistant Professor,

School of Electronics Engineering,

VIT University,Vellore,India.

e-mail : [5]vidhyapathi.cm@vit.ac.in ; [8]satheeshkumar.s@vit.ac.in

[9] Assistant Professor, School of Artificial Intelligence, Amrita Vishwa vidhyapeetham, Bangaluru,India.

e-mail : [9]cs_vasavi@blr.amrita.edu

[10] Professor,Communication and networks department,College of Engineering,Prince Sultan UniversityRiyadh, Saudi Arabia.

e-mail : [10]usubramaniam@psu.edu.sa

[1*]Corresponding Author :sundar.s@vit.ac.in

Abstract— More than half of all people on the planet voluntarily keep pets as companions, but these animals are starting to strain their owners. While feeding their pets, owners experience a lot of stress**.** One of the pet feeders that may be managed online through a mobile application is the IoT pet feeder. This paper details an IoT based monitoring system which automates the process of detecting and replacing dirty water in a pet's bowl in a cheap manner. It also describes methods of tracking the overall water consumption of the pet via the system and monitoring the pet's health via detection of consumption anomalies.

Keywords- Internet of Things, Smart Electronic Systems, Sensors, Microcontrollers, EdgeLearning, TinyML

I. Introduction

Pets, just like human children, require special treatment and attention. This task is not as straight forward as it once was due to our busy lifestyles and stressful schedules. In metropolitan areas, parents soften have to leave their pets unattended for extended periods of time. Our proposal draws from the concept s of IoT and smart systems to automate part of a parent's workloads in such situations.

Most pets have a dedicated water bowl. The water here accumulates dust and other physical impurities over time. Thus, it is essential to replace this water regularly. The proposed system will automate this process by regularly measuring the turbidity values

Article Received: *17 August 2023* ***Revised:*** *20 September 2023* ***Accepted:*** *10 October 2023*

and raising an alert or replacing the water when the value crosses a threshold.

At the same time, the system will also monitor the water consumption patterns of the pet. It is often difficult to gauge the health conditions of a pet directly. However, pets tend to alter their water and food consumption patterns on feeling any kind of discomfort. Thus, by comparing the water consumed against historical trends under similar conditions, it is possible to draw an inference towards the pet's health. Though not prescriptive in nature, it can be indicative of something being wrong.

This paper first explores the hardware used in the system in section II. It lays a foundation for the selection of certain components and how they work before demonstrating their interconnected working. It then discusses the prediction algorithm used to determine the water consumption trends in detail in section III. Next, in section IV, the case of edge deployment of the algorithm on the Raspberry PI is explored along with Tiny ML. Finally, the paper shows how IFTTT is used as a web-broker for end-user communication in section V.

II. THE HARDWARE

A. The Microcontroller

The system was first prototyped with an Arduino UnoRev3 which uses the ATmega328P microchip. Though satisfactory in the collecting and processing of data, we felt it did not meet standards with regards to the scalability of the project, especially with edge learning models being used to track and predict the water consumption patterns. Our final choice was the Raspberry Pi Pico W. Though comparable to the Uno in terms of form factor, price and power consumption, its Dual-core Arm Cortex M0+ processor and the RP2040 microcontroller chip ensure it can run heavy duty edge learning models. It also has an on-board single-band 2.4GHz wireless interface (802.11n) using the In fine on CYW43439 which ensures wireless connectivity without the need for a separate module. We felt the Pico W matched our IoT system requirements adequately at no extra cost.

B. The Sensors

The proposed system measures the water turbidity using a SKU: SEN0189 analog gravity turbidity sensor from DF Robot. It uses light to detect suspended particles in water by measuring the light transmittance and scattering rate (nephelometry), which changes with the amount of Total Suspended Solids (TSS) in water [1][2]. As the TSS increases, the liquid turbidity level increases. The final turbidity is calculated as a function of the output voltage and the turbidity (in Nephelometric Units or NTUs) as shown in figure1.

The system also uses a generic DHT11–Temperature and Humidity Sensor and a HC-SR04 ultrasonic distance sensor. The temperature and humidity on a particular day can affect how much water is being drunk. Thus, these factors are taken into consideration to calculate the water consumption patterns. The ultrasonic sensor, fixed at a particular height perpendicularly above the surface of the water bowl is used to determine the level of water in the bowl [3] [4] [5] [6].

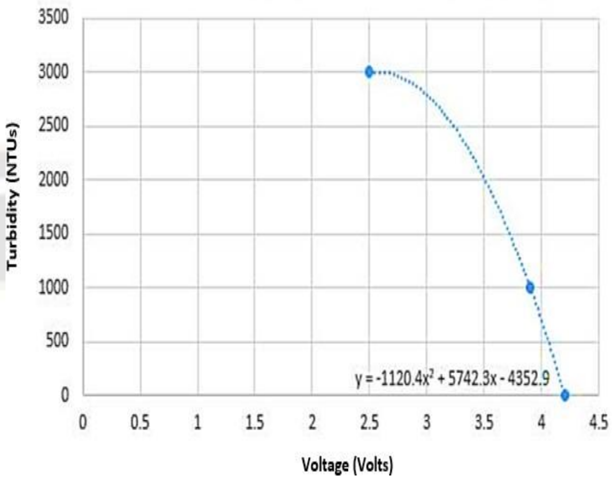
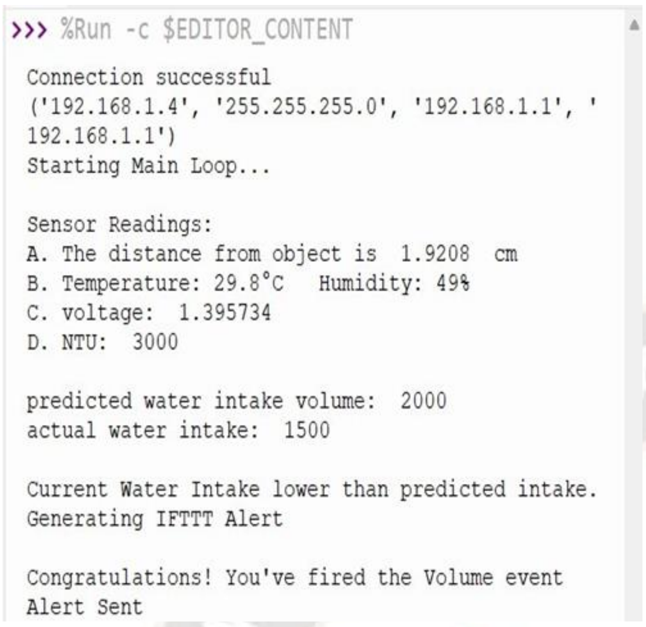


Figure 1. Voltage vs. Turbidity (in NTUs).

C. Working

The workflow of this system is mentioned in Fig 2

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 17 August 2023 **Revised:** 20 September 2023 **Accepted:** 10 October 2023*



```
>>> %Run -c $EDITOR_CONTENT
  Connection successful
  ('192.168.1.4', '255.255.255.0', '192.168.1.1', '
  192.168.1.1')
  Starting Main Loop...

  Sensor Readings:
  A. The distance from object is  1.9208  cm
  B. Temperature: 29.8°C   Humidity: 49%
  C. voltage:  1.395734
  D. NTU:  3000

  predicted water intake volume:  2000
  actual water intake:  1500

  Current Water Intake lower than predicted intake.
  Generating IFTTT Alert

  Congratulations! You've fired the Volume event
  Alert Sent
```

Figure 2. Hardware process flow chart.

At the start of an interval, the system measures the time, current water level and the turbidity. If the turbidity is too high, a valve opens to replace the entirety of the water. If the volume is too low, a certain additional volume is dispensed and the entire volume is noted. If it is the end of the day, all the dispensed volumes are added up and along with the temperature and humidity for that day, it is passed to the

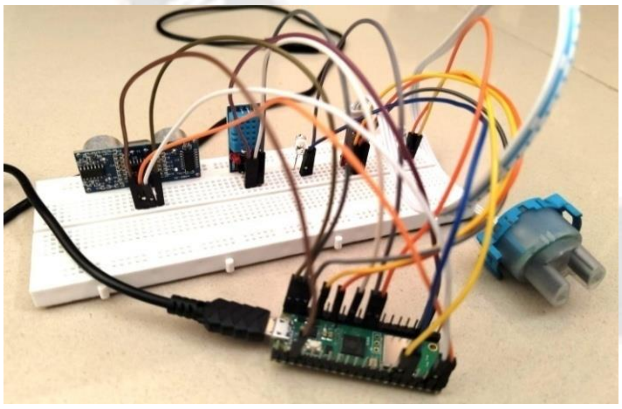


Figure 3. Hardware prototype made using the Raspberry Pi Pico W.

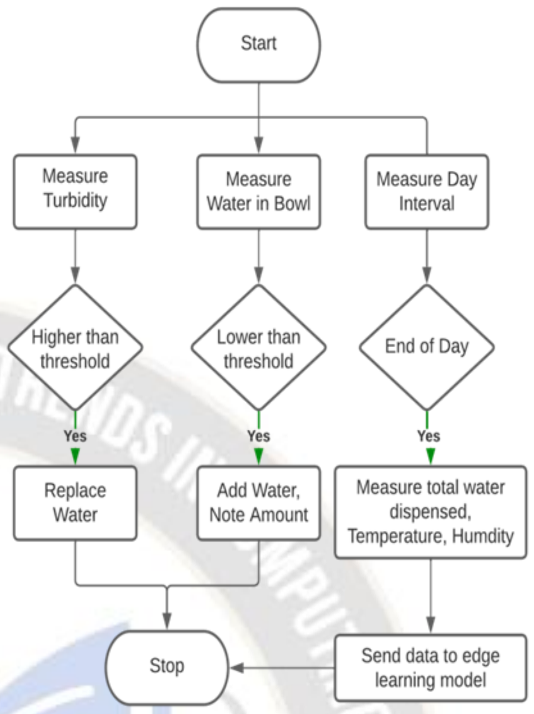


Figure 4. Hardware output on the Micro Python console.

edge learning model. The model adds this to its database, and checks compare the total volume with a prediction value it comes up with using the day's temperature, humidity, and historical trends. If this comparison shows the pet drank too much or too little water in comparison to its predicted value, an alert is thrown to the user. Figure2 shows a flowchart detailing this process. Figure 3 shows the hardware prototype made using the Raspberry Pi Pico and the aforementioned sensors. Figure 4 shows the sample output of the hardware model on a Micro Python console with the collected data values and action items.

III. THE PREDICTION MODEL

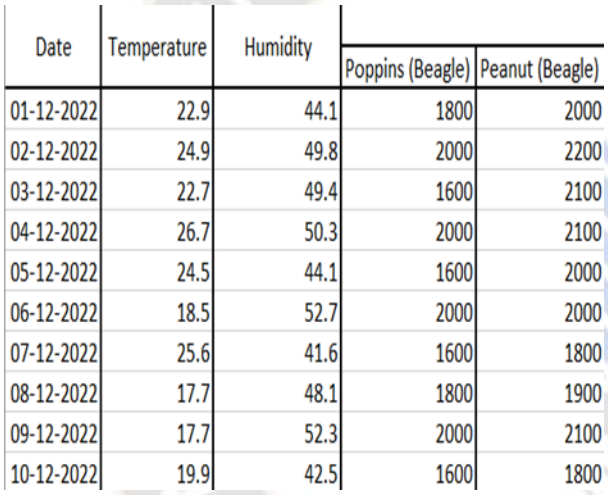
The prediction model shows the current day's temperature, humidity and total volume of water consumed. The days temperature and humidity are used to calculate the predicted volume using historical trends. If the predicted volume is higher than the actual volume, it means the pet drank less water than usual. This throws an alert. The volume,

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 17 August 2023 **Revised:** 20 September 2023 **Accepted:** 10 October 2023*

temperature and humidity is appended to the database. This creates a system which gets more accurate overtime.

A. The Database

The Database contains the historical figures for temperatures, humidity and total volume of water consumed recorded daily as shown in figure 5. It contains data for over 20 pets of different species collected over 2 months to avoid bias. This data is aggregated and further processed to remove extreme outliers and missing fields before being sent to the prediction algorithm.



Date	Temperature	Humidity		
			Poppins (Beagle)	Peanut (Beagle)
01-12-2022	22.9	44.1	1800	2000
02-12-2022	24.9	49.8	2000	2200
03-12-2022	22.7	49.4	1600	2100
04-12-2022	26.7	50.3	2000	2100
05-12-2022	24.5	44.1	1600	2000
06-12-2022	18.5	52.7	2000	2000
07-12-2022	25.6	41.6	1600	1800
08-12-2022	17.7	48.1	1800	1900
09-12-2022	17.7	52.3	2000	2100
10-12-2022	19.9	42.5	1600	1800

Figure 5. Small portion of the collected database.

B. The ML Algorithm

Technically, our model needs to find the relationship between 3 variables. Temperature and humidity can be considered as independent variables for simplicity's sake, and the water volume can be considered as the dependent variable or the target label. This model can be either hosted in the cloud, with the system sending data to it, or it can be run at the edge level. With the rising alarms over data security and privacy and the benefits of a self-contained system, we choose to deploy our model on our edge device itself. This meant there was an increased importance placed on speed and simplicity for the model. Taking all these factors into consideration, a regression model seemed to be the most optimal. It is fast, and the use case does not demand a more powerful algorithm [7]. The outputs of our regression model are shown in figure 6, with the total volume in blue and the predicted volume in red.

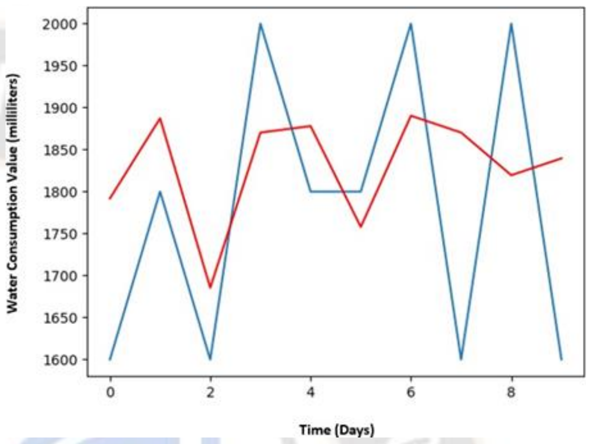


Figure 6. Regression model results.

For testing purposes, we compared this model against models based on the Random Forest Classifier, K-Nearest Neighbors and Support Vector Machine algorithms trained on the same data. The regression models gave better accuracy and over root mean squared errors (RMSE) as well as lower execution times. As such, the regression-based algorithm was finalized [7].

IV. Edge Development

The prediction model discussed earlier was coded in python using frameworks from the scikit-learn machine learning library. The Raspberry Pi on the other hand was programmed using Micro Python, a lean and efficient implementation of the Python3 programming language that includes a small subset of the Python standard library and is optimized to run on microcontrollers and in constrained environments As such, Micro Python supports neither scikit-learn nor the dependencies scikit-learn is built on out of the box. Though a few sub-libraries of scikit-learn have been ported over to Micro Python, these are not optimized.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 17 August 2023 **Revised:** 20 September 2023 **Accepted:** 10 October 2023*

A new field called Tiny ML has come up specifically to deal with these issues. Tiny ML is a field of study in Machine Learning and Embedded Systems that explores the types of models you can run on small, low-powered devices like microcontrollers. It enables low-latency, low power and low bandwidth model inference at edge devices. While a standard consumer Central Processing Unit(CPU) consume between 65 watts and 85 watts and standard consumer Graphics Processing Unit (GPU) consumes anywhere between 200 watts to 500 watts, a typical microcontroller consumes power in the order of milli watts or microwatts. That is around a thousand times less power consumption. This low power consumption enables the Tiny ML devices to run unplugged on batteries for weeks, months, and in some cases, even years, while running ML applications on edge [8].

A. *Edge Impulse*

Edge Impulse is a cloud-based Machine Learning Operations (ML Ops) platform for developing embedded and edge machine learning (Tiny ML) systems that can be deployed to a wide range of hardware targets [9]. The plat form can build a machine learning (ML) impulse (an impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data) tuned to a particular application which can then be deployed to the microcontroller as a C++ library or an executable, thus bypassing the restrictions of Micro Python.

B. *Building the impulse*

Creating an Impulse involves the following steps:
- Data Acquisition
- Impulse Design
  - – Impulse creation
  - – Raw data analysis
  - – ML model analysis
- Mode testing and versioning
- Deployment

Data Acquisition can be done by either live collection, which needs the edge device to be flashed with the edge impulse firmware, or by importing a preexisting database. In our case, we imported the database described earlier as a .CSV file. The platform also helps in visualizing and pre-processing the data, as shown in figure7, with the temperature waveform in orange and humidity wave form in green.

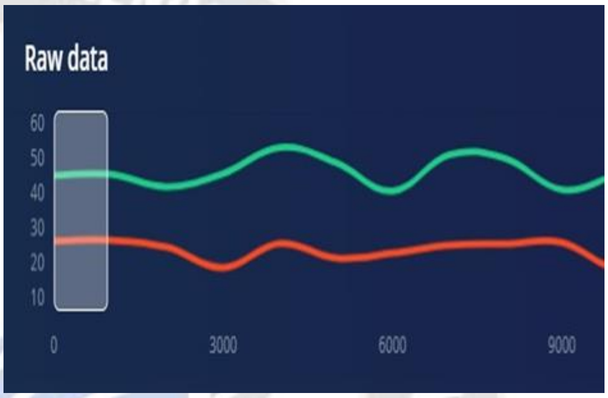
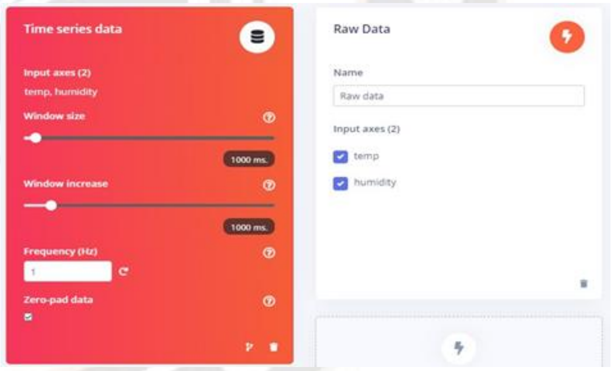


Figure 7. Visualizing the data on the Edge Impulse platform.



Input axes (2)
temp, humidity
Window size
1000 ms.
Window increase
1000 ms.
Frequency (Hz)
1
Zero-pad data ✓

Name
Raw data
Input axes (2)
✓ temp
✓ humidity

Figure 8. Data acquisition and processing blocks of the impulse.

The next step involves creating the impulse with this data. Here we established the data pipeline and the regression block. Edge Impulse offers pre-built blocks of algorithms, but we built one using our own regression model discussed in section III. III. Figure 8 shows the initial blocks of the impulse which convert the imported data into a time-series data and extract its features. Figure 9 shows the custom regression block which receives the features from the time series and data and returns a single scalar value to the final output block.

Once the impulse was established, we trained it with the loaded database. We were able to achieve an accuracy of 95% over multiple trials. The results are shown in figure 10 in the form of a scatterplot. The X axis represents the date of the features taken to predict the value whereas the Y axis represents the magnitude of the predicted value. Green dots show a correct regression prediction whereas red dots indicate a wrong prediction.

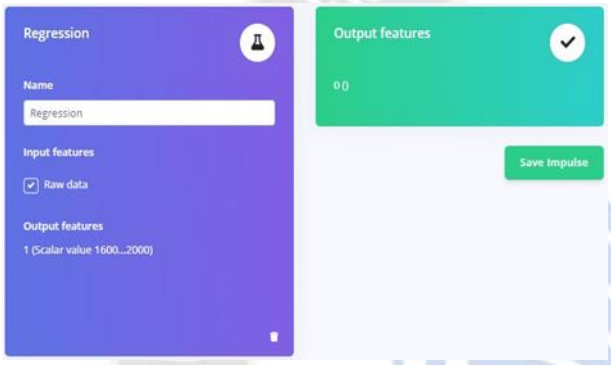


Figure 9. Custom regression and final output blocks of the impulse.

We finally built the firmware for this impulse for the RP2040chip used in our Raspberry Pi Pico. This firmware executable bundles both the database and the pre-trained impulse into one package. Once flashed onto the Pico, the Micro Python code controlling the Pico acts as an inference code. It calls the executable and passes the required values as function arguments. The executable returns the predicted value. This also means that the Pico does not need a separate copy of the database, thus saving valuable memory space. The build results for the executable are shown in figure 11. The final executable would need 1.5 Kilobytes of Random Access Memory (RAM) and 9, 4 Kilobytes of flash memory while being run. These values are very low, thus making it possible for it to be deployed to any edge device.

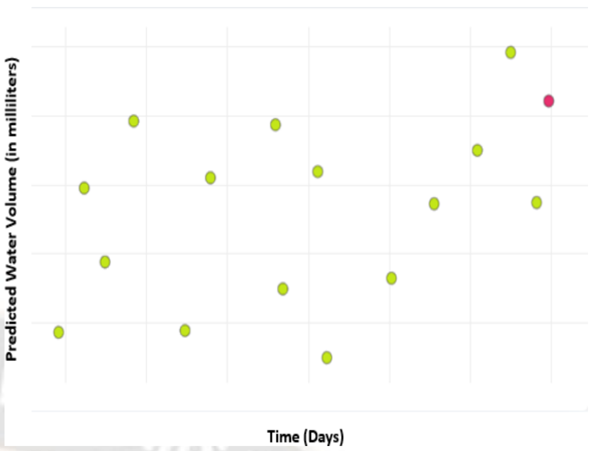


Figure 10. Regression impulse results.

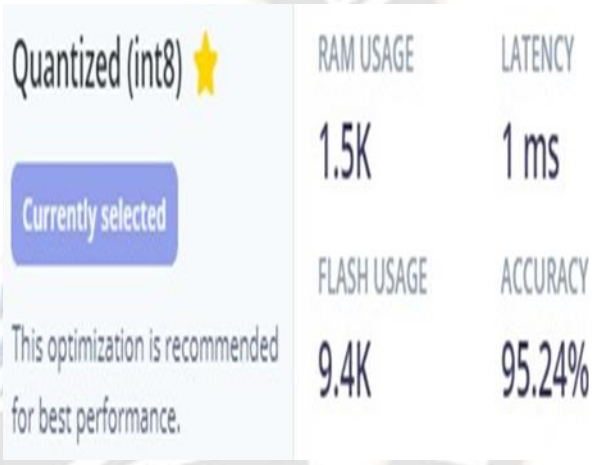


Figure 11. Firmware builds results.

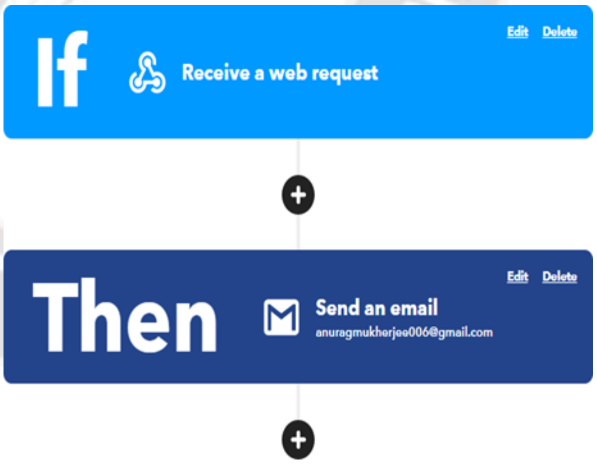
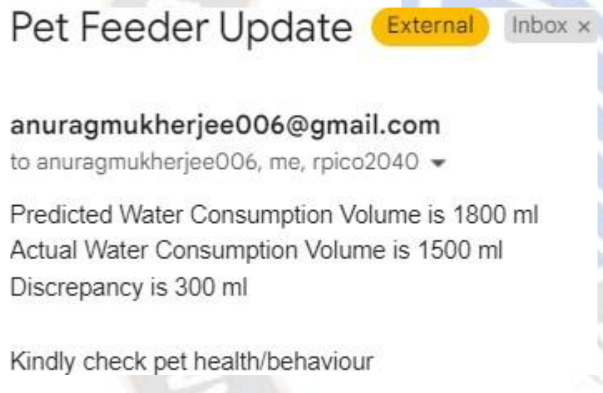


Figure 12. IFTTT applet

V. END USER INTERFACE

We used IFTTT, a digital automation and cross platform connectivity service to publish alerts for the end user [10]. The Micro Python code compares the total value to the predicted value coming from the impulse model. If any discrepancies are found, it packages the required discrepancies and values as a JSON payload and pushes it to the IFTTT broker. On the IFTTT server, an applet with a web-hook waits for any kind of communication. When it receives the JSON payload, it unpacks the values and sends an e-mail alert to a predefined address with these values. The applet can be configured to work with various end user services, including pushing messages to android mobiles and even social media platforms depending on the user's preference. The applet showing the IF and THEN conditions is shown in figure 12 and the sample mail alert received from the system is shown in figure13.


Pet Feeder Update External Inbox ×

anuragmukherjee006@gmail.com
to anuragmukherjee006, me, rpico2040

Predicted Water Consumption Volume is 1800 ml
Actual Water Consumption Volume is 1500 ml
Discrepancy is 300 ml

Kindly check pet health/behaviour

Figure 13. Sample mail alert received from system.

VI. CONCLUSION

The proposed system will considerably ease a pet parent's workload by using automation and IoT to resolve a real-world issue. The system is also considerably cheaper than other similar solutions available on the market. It uses an entry level microcontroller with everyday sensors. As such, it is also easy to modify and maintain. The successful use of edge learning over the traditional cloud-based models used in similar IoT systems further demonstrated the potential of Tiny ML as a concept which can combat data security and privacy issues.

REFERENCES

[1] H. Hendri, S. Enggari, Mardison, M. R. Putra, and L. N. Rani, "Automatic system to fish feeder and water turbidity detector using Arduino Mega," Journal o Physics: Conference Series, vol .1339, no.1, p.012013, 2019.

[2] Y. K. Taru and A. Karwankar, "Water monitoring system using Arduino with lab view, " 2017 International Conference on Computing Methodologies and Communication (ICCMC), 2017.

[3] S. O. Osman, M. Z. Mohamed, A. M. Suliman, and A. A. Mohammed, "Design and implementation of a low-cost real-time in-situ drinking water quality monitoring system using Arduino," 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), 2018.

[4] G. Vidya Shree and Sharanabasappa, "Non-contact water level monitoring using lab view with Arduino and ultrasonic sensor," Third International Conference on Current Trends in Engineering Science and Technology ICCTEST-2017, 2017.

[5] T. Malche and P. Maheshwary, "Internet of things (IOT) based Water Level Monitoring System for smart village," Advances in Intelligent Systems and Computing,pp.305–312,2017.

[6] A. A .Ismail, M. A. Azizi, and A. Zariman, "Smart water level indicator, "International Journal of Recent Technology and Applied Science, vol.2, no.1, pp. 48–58, 2020.

[7] E.Walker, "Regression modeling strategies, "Techno metrics, vol.45, no .2, pp.170–170, 2003.

[8] D. L. Dutta and S. Bharali, "Tiny ML meets IOT: A comprehensive survey, "Internet of Things, vol.16,p.100461, 2021.

[9] Shawn Hymel, Colby Banbury, Daniel Situnayake, Alex Elium, Carl Ward, Mat Kelcey, Mathijs Baaijens, Mateusz Majchrzycki, Jenny Plunkett, David Tischler, Alessandro Grande, Louis Moreau, Dmitry Maslov, Artie Beavis, Jan Jongboom, and Vijay Janapa Reddi, "Edge Impulse: An ML Ops Platform for Tiny Machine Learning," 2022.

[10] S. Ovadia, "Automate the internet with 'if this then that' (IFTTT),"Behavioral and Social Sciences Librarian, vol. 33, no. 4, pp. 208–211, 2014.