An ML-based water flow prediction model with pressure sensor

Maulesh Gandhi, A. Mathew, S. Chaudhari, R. Shaik, A. Vattem
International Institute of Information Technology Hyderabad (IIIT-H), India
Email: {maulesh.gandhi, ajai.mathew}@research.iiit.ac.in,
{sachin.chaudhari, rehana.s}@iiit.ac.in, anuradha.vattem@research.iiit.ac.in

***Abstract*—This study presents an Internet of Things (IoT)-based water flow prediction model that utilizes machine learning techniques to analyze water flow through pipes. The model incorporates an ESP-32 microcontroller, a Danfoss MBS 3000 pressure sensor, and a flow meter, deployed at three locations for a two-month period to collect data. Preprocessing steps were performed to enhance the dataset quality, and the relationship between pressure and flow rate was evaluated using Pearson's correlation coefficient.**

The results demonstrate the effectiveness and reliability of the proposed model in accurately predicting water flow. By establishing a model that captures the relationship between pressure and flow rate, the need for a flow meter in the setup can be eliminated. The low-cost and easy implementation of the system make it suitable for widespread adoption in residential areas, facilitating efficient water management and leak detection efforts. The model offers advantages such as cost-effectiveness, real-time data analysis, and efficient water management.

In conclusion, this IoT-based water flow prediction model presents a promising solution for optimizing water distribution and reducing water wastage.

Keywords: IoT, machine learning, water flow prediction, pressure sensor, ESP-32.

I. Introduction

Access to good quality water for drinking and residential purposes is a growing concern due to its increasing scarcity. Traditional methods for measuring water consumption and billing involve using analog water meters, which rely on mechanical turbine-based volume measurement [1], [2]. While this method is reliable for measuring water, it has limitations in terms of human dependency for reading collection and the inability to detect or predict leaks. Various digital solutions have emerged to automate meter reading through cloud servers, but they often retain the same measuring method as analog meters. Recently, there has been a development in retrofitting existing analog meters with low-cost solutions that utilize machine learning (ML) algorithms and image processing, addressing the drawbacks of traditional approaches [3]. However, challenges remain in terms of dependency on the existing meter and its dial orientation.

To overcome these challenges, this paper proposes an ML and IoT-based low-cost flow prediction hardware that enhances user-friendliness and affordability. Our model offers several advantages over existing methods by providing real-time data on water consumption patterns, being cost-effective, and easy to use. The integration of IoT enables efficient data collection and validation, which is crucial for obtaining accurate readings and training the ML algorithm. By capturing data from distributed pipelines and aggregating it at a central point, IoT plays a significant role in data collection [4], [5]. Existing research primarily focuses on leak detection and water wastage prevention, utilizing approaches such as sensing pipe vibrations, identifying leak locations, and detecting pressure variations [1], [2], [4], [6]. Some studies have explored using existing analog meters for digital data collection rather than replacing them entirely [3]. However, limited research has been conducted on measuring water consumption, as the existing measurement process has been deemed efficient and the cost value for consumed water is considered negligible. However, as government organizations increasingly seek to identify non-revenue water and its sources, the need for effective digitization becomes evident.

In this paper, we present an IoT-based low-cost sensor node designed and deployed in the field across three different locations on water pipelines. The hardware employs minimal components and offers reliable performance over an extended period. Data is collected from water pumping lines over a period of three months, serving a campus with over 3000 residents. An ML algorithm analyzes the data and models the relationship between input pressure and predicted flow rate. The proposed approach aims to use the trained model to predict water flow by monitoring the pressure inside the pipe.

The advantages of our solution lie in its simplicity and cost-effectiveness, making it suitable for implementation in residential areas where there is no need to add costly new digital meters as part of digitization efforts. Customers can gain insights into the timing and effectiveness of water distribution, while distributing agencies can assess pressure on different lines and route water more effectively.

The rest of the paper is organized as follows. Section II provides details on the development of IoT hardware and data collection over a two-month period. Section III presents the data analysis using ML tools. The results are discussed in Section IV, and finally, Section V concludes the paper.

II. Hardware Description

A. Hardware setup description

The block architecture of the node developed at IIIT-H is shown in Fig. 1. Every node consists of a pressure sensor, a digital flow meter, and an ESP32 for measuring the pressure

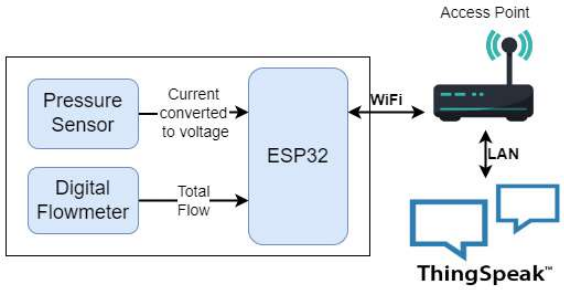


Fig. 1: Block architecture of the node

and flow rate of water going through pipelines. An additional AI-PRI digital pressure indicator by ACE instruments is deployed at one of the deployment locations. The purpose of this pressure indicator is to estimate the pressure values more accurately and validate our pressure estimations. The pressure sensor gives a current output of 4-20 mA based on the pressure inside the pipes. This is converted to voltage using a resistor and then read by the ESP32. The readings from the flow meter and pressure indicator are read using MODBUS communication. We use an external adaptor to provide the 12V required for the pressure sensor and a regulator to convert it to the 5V required by the ESP32. The software code which runs in the ESP-32 updates the values to the Thingspeak cloud server using Wi-Fi.

B. Pressure transmitter description

The Danfoss MBS 3000 pressure transmitter [7] is an electronic sensor that measures pressure in various industrial applications. It is based on piezoresistive technology, which means that it uses a particular type of sensor that changes its electrical resistance in response to changes in pressure. The MBS 3000 is designed to measure pressure up to 600 bar and is available in various output signals such as 4-20 mA, 0-10 VDC, or 0-5 VDC.

The MBS 3000 pressure transmitter consists of a sensing element, a signal conditioning circuit, and microprocessor-based digital circuitry. The sensing element is a thin silicon diaphragm that deforms in response to pressure changes. This deformation causes a change in the resistance of the piezoresistive material, which is then converted into an electrical signal by the signal conditioning circuit. The version of MBS 3000 we have opted for has a pressure range of 0-16 bar and gives an output current of 4-20 mA for the pressure range. The current output feature enables the sensor to be kept significantly apart from the IoT node, which means the electronic circuit need not be near the pipelines. We use a simple resistor connection to convert this current to a voltage level acceptable by the ESP-32. The performance specifications of the sensor are provided in Table I.

C. Data collection

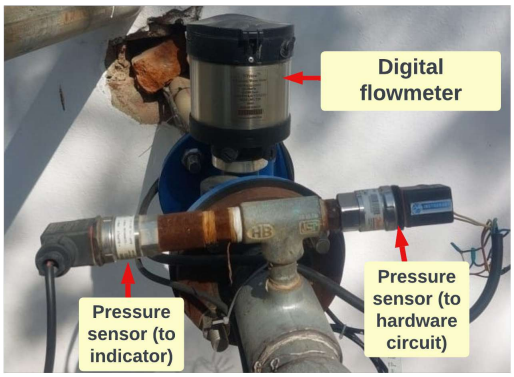
With the help of the above-described hardware setup, three nodes were deployed at different locations: one at the pump

TABLE I: Pressure sensor specifications

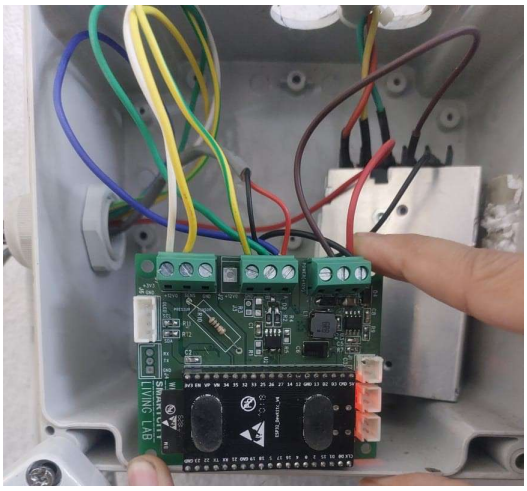
Accuracy	$\leq \pm$ 0.5% FS (typ.) $\leq \pm$ 1% FS (max.)
Non-linearity	$\leq \pm$ 0.2% FS
Hysteresis and repeatability	$\leq \pm$ 0.1% FS
Thermal zero-point shift	$\leq \pm$ 0.1% FS / 10K (typ.) $\leq \pm$ 0.2% FS / 10K (max.)
Thermal sensitivity (span) shift	$\leq \pm$ 0.1% FS / 10K (typ.) $\leq \pm$ 0.2% FS / 10K (max.)
Response time (liquids with viscosity $<$ 100 cSt)	$<$ 4 ms
Response time (air and gases, MBS 3050)	$<$ 35 ms
Overload pressure (static)	6 × FS (max. 1500 bar)
Burst pressure	6 × FS (max. 2000 bar)
Power-up time	$<$ 50 ms
Durability (P : 10–90%FS)	$>$ 10 × 106 cycle

Fig. 2: Locations of deployed nodes

house pumping water for domestic usage to most of the college hostels (PH0100), and two at the output of another pump house pumping drinking water to separate faculty quarters (PH0302 and PH0303). The locations of deployed nodes are shown in Fig. 2. Data was collected over two months for the three nodes. Over 26,000 data points were collected for each node. All the nodes have pressure sensors and water flow meters. Only the PH0100 node has an additional AI-PRI digital pressure indicator which converts the pressure sensor readings directly to pressure values with an accuracy of 0.25 % of F.S. and gives it to the microcontroller. The indicator has been calibrated using an advanced pressure factor marker to estimate pressure more accurately and advantageously. Since these values are more accurate than the measured pressure values, which have an accuracy of 0.5 % of F.S., they are used to validate them. Fig. 3 shows the PH0100 node at which the pressure indicator has also been deployed.



(a) Pressure sensor and digital watermeter



(b) Hardware circuit

Fig. 3: Subfigures of Pressure sensor and digital flow meter and Hardware circuit

III. Tools used for analysis

A. Preprocessing

The dataset collected from the deployed nodes underwent several preprocessing steps to enhance the accuracy and reliability of the models. The preprocessing steps included:

1) Calculation of Flow Rate: The total flow readings and corresponding timestamps were used to calculate the flow rate at each time point.
2) Outlier Removal: K-means clustering was applied to identify outliers in the dataset. The optimal number of clusters was determined based on evaluation metrics such as the Silhouette coefficient, Calinski-Harabasz score, and Davies-Bouldin score. Outliers were defined as data points that deviated significantly from the median of their respective clusters. These outliers were replaced with the median values of their clusters.
3) Data Smoothing: A moving average technique with a fixed window length of 5 was applied to smooth the data and reduce noise. This helped in removing short-term fluctuations and highlighting the underlying trends in the dataset.

These preprocessing steps were crucial in improving the dataset's quality by reducing noise, removing anomalous data points, and capturing the underlying patterns more accurately.

B. Correlation coefficient

After preprocessing the data, we calculated Pearson's correlation coefficient between our variables. Pearson's correlation coefficient, denoted by r, measures the strength of a linear association between two variables. A coefficient close to 1 or -1 suggests a strong positive or negative linear relationship, respectively, while a coefficient close to 0 indicates no significant linear relationship. Pearson's correlation coefficient detects linear relationships between variables but may not be as effective in detecting non-linear relationships.

C. Training ML models

The ML models used in this study were trained on data collected from the three deployed nodes. The dataset included pressure sensor readings and corresponding flow rates measured by the water flow meter. The data collection process spanned two months to capture various flow patterns and consumption behaviours. The dataset underwent preprocessing steps before training the ML models to improve accuracy and remove outliers. Three different ML algorithms were employed for training:

1) Linear regression: Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to the observed data. It served as a baseline model for comparison with more complex algorithms. The linear regression model aimed to capture the linear relationship between the input pressure readings and the predicted flow rate.
2) Support vector regression (SVR): SVR is a robust regression algorithm that captures complex relationships between input features and the target variable. It leverages support vector machines to approximate the mapping function. SVR can effectively model non-linear data patterns, and it was employed in this study to capture non-linear relationships between pressure sensor readings and flow rates.
3) Convolutional neural network (CNN): CNNs are deep learning models widely used for image and sequence analysis tasks. In this study, a CNN model was utilized to leverage the spatial and temporal patterns in the dataset. Given the high correlation between pressure values and flow rate, our study employed a basic CNN architecture consisting of two convolutional layers, Rectified Linear Unit (ReLU) activation, and one final dense layer. The final dense layer consists of a single neuron since the model performs a regression task (predicting a single continuous value). The CNN models were trained for 100 epochs using the data from each node and evaluated using the testing set.

The training set was used to train each model, and the testing set was used to evaluate their performance. The models were assessed based on various evaluation metrics, which provided insights into the accuracy and predictive capabilities of the trained models.

By employing a combination of linear regression, SVR, and CNN models, the study aimed to capture the relationships between pressure sensor readings and flow rates from different perspectives, allowing for more accurate and robust predictions.

D. Testing

After training the machine learning model on the labeled dataset, we proceeded with testing its performance using k-fold validation. K-fold validation is a technique commonly used to assess the generalization ability of a model by dividing the dataset into k equally-sized folds or subsets.

In our experiments, we performed 10-fold validation, where the dataset was randomly partitioned into 10 subsets of approximately equal size. We iteratively trained the model on 9 folds and evaluated its performance on the remaining fold. This process was repeated 10 times, each time using a different fold as the test set, and the performance metrics were averaged across all iterations.

By employing k-fold validation, we obtained more accurate and reliable estimates of the model's performance. Here are some benefits of using k-fold validation:

1) Reduced bias: K-fold validation helps mitigate the impact of a particular training-test split by averaging the performance across multiple folds. This reduces the potential bias introduced by a single training-test partition.
2) Improved generalization: The performance metrics obtained from k-fold validation are generally more representative of the model's ability to generalize to unseen data. This is because the model is trained and evaluated on different subsets of the dataset, providing a more robust assessment.
3) Better parameter tuning: K-fold validation enables more effective hyperparameter tuning. By evaluating the model's performance across multiple folds, we can identify parameter settings that consistently yield optimal results.

During the k-fold validation process, we recorded root mean squared error (RMSE) and accuracy or coefficient of determination (R^2) scores to comprehensively evaluate the model's effectiveness in classifying the dataset's instances. These metrics were computed for each fold and then averaged to provide an overall assessment of the model's performance.

E. Evaluation metrics

Our machine learning models' performances were evaluated using root mean squared error (RMSE) and accuracy/coefficient of determination (R^2). The RMSE measures the root average squared difference between the predicted and actual flow rate values (unit is Kl per hour). In contrast, R^2 measures the proportion of variance in the dependent variable that is predictable from the independent variable. The coefficient of determination R^2 is defined as $(1 - \frac{u}{v})$, where u is the residual sum of squares $\sum(y_{true} - y_{pred})^2$ and v is the total sum of squares $\sum(y_{true} - \bar{y}_{true})^2$.

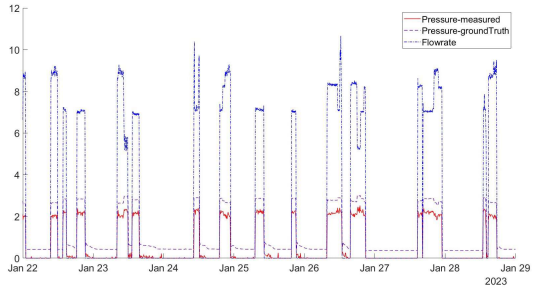


Fig. 4: Processed Data

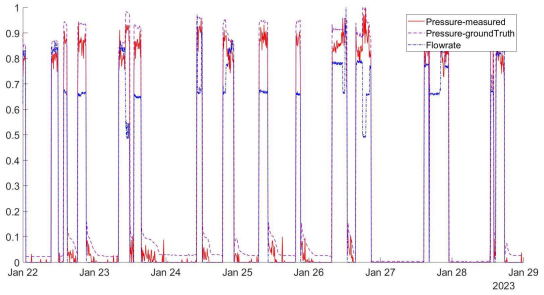


Fig. 5: Normalized Data

IV. Results and Observations

A. Time Series Data

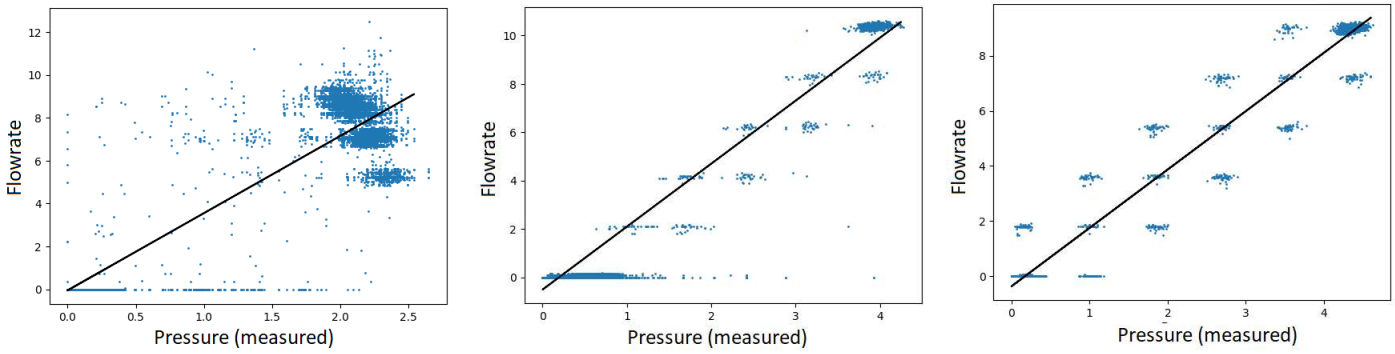
Fig. 4 shows the time-series plot over 4 days obtained after applying our preprocessing steps to the original data. Fig. 5 shows the same data after normalizing the processed data for better visualization by scaling it to a range of 0-1. Both the plots show a high correlation between pressure (ground truth), pressure (measured), and flow rate, which can also be observed from the correlation coefficients of all nodes, given in Table II. This is the basis for the proposed algorithm. Also, due to the high correlation observed between the measured pressure values and the ground truth pressure values obtained from the pressure indicator, we can concur that there is no need for data calibration.

B. Machine Learning Results

After preprocessing the data and calculating the correlation coefficient between our variables, we applied linear regression

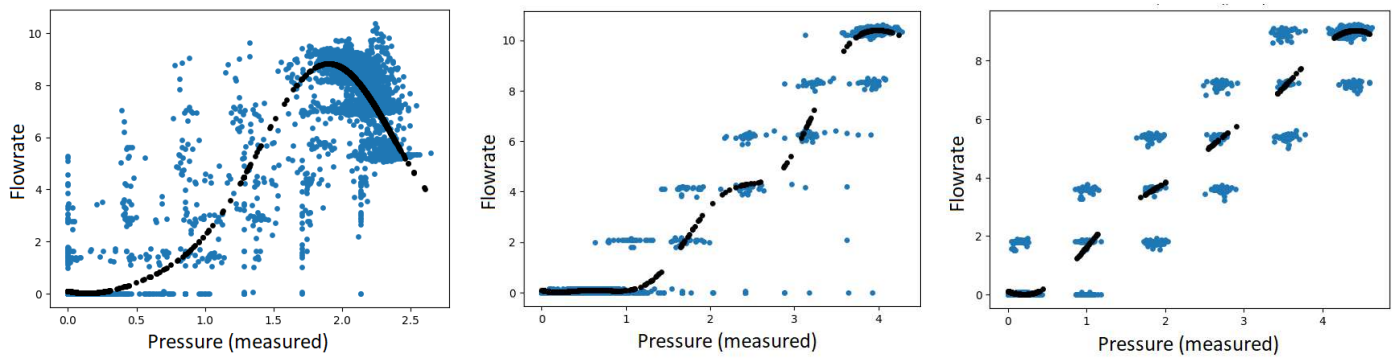
TABLE II: Correlation coefficients between pressure and flow for different locations

	PH0100	**PH0302**	**PH0303**
Pressure (measured) and flowrate	0.970	0.973	0.990
Pressure (measured) and pressure (ground truth)	0.993	-	-



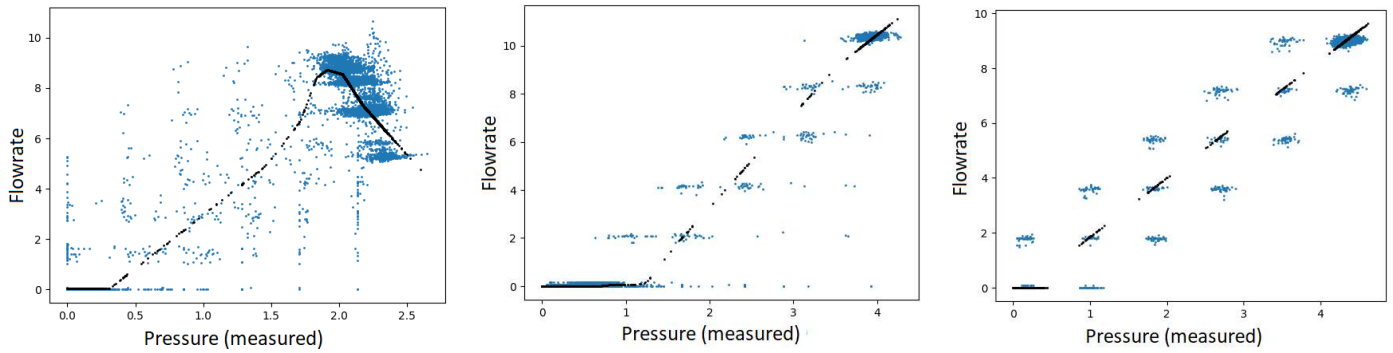
(a) PH0100 (b) PH0302 (c) PH0303

Fig. 6: Linear regression scatterplots



(a) PH0100 (b) PH0302 (c) PH0303

Fig. 7: SVR scatterplots



(a) PH0100 (b) PH0302 (c) PH0303

Fig. 8: CNN scatterplots

TABLE III: Accuracies of ML models trained

Model	PH0100	PH0302	PH0303
LR	0.942	0.946	0.980
SVR	0.969	0.981	0.984
CNN	0.974	0.984	0.984

TABLE IV: RMSE (in Kl) values of ML models trained

Model	PH0100	PH0302	PH0303
LR	0.873	0.450	0.302
SVR	0.635	0.253	0.269
CNN	0.579	0.232	0.273

to model the relationship between the measured pressure and the predicted flow rate. The results obtained after training the model are presented in Fig. 6. The black lines represent the fitted line based on our data. The measured pressure data from the other two nodes was also used to train models for respective nodes. All the nodes gave an accuracy of above 90%, which validates our attempt at predicting flow rates.

The SVR models demonstrated superior performance to linear regression, achieving higher accuracies and lower RMSE values. The SVR results indicate that the predicted flow rates based on pressure readings are more accurate and closer to the actual flow rates. To visualize the performance of the SVR models, we generated scatterplots showing the predicted flow rates versus the actual flow rates based on the testing set. The scatterplots for trained models can be seen in Fig. 7. The scatterplots illustrate the relationship between the predicted flow rates and the actual flow rates. Ideally, the points on the scatterplot should form a linear pattern, indicating a strong correlation between the predicted and actual values. A more scattered pattern suggests a less accurate prediction. Based on the scatterplots, we can observe the performance of the SVR models and visually assess their accuracy in capturing the underlying relationships.

To visually assess the performance of the CNN models, we generated scatterplots showing the predicted flow rates versus the actual flow rates based on the testing set. The scatterplots for each node can be seen in Fig. 8. By examining the scatterplots, we can assess the accuracy of the CNN models in capturing the underlying patterns and making accurate predictions. A tight cluster of points around the diagonal line indicates a strong correlation between the predicted and actual values. The CNN models outperformed linear regression and SVR, yielding higher accuracies and lower RMSE values. These results indicate that CNNs effectively capture the complex relationships between pressure values and flow rate, resulting in more accurate flow rate predictions.

The accuracies and RMSE errors obtained for trained ML models of all nodes are shown in Tables III and IV respectively.

V. Conclusion

In conclusion, this study introduces an IoT-based water flow prediction model that utilizes machine learning algorithms to analyze water flow through pipes. The model's hardware setup, including an ESP-32 microcontroller, a Danfoss MBS 3000 pressure sensor, and a flow meter, was deployed at three distinct locations over a two-month period to gather data. Through preprocessing steps and the application of Pearson's correlation coefficient, the relationship between pressure and flow rate was assessed.

The findings of this study demonstrate the effectiveness and reliability of the proposed model in accurately predicting water flow. By establishing a robust model that captures the correlation between pressure and flow rate, the reliance on a flow meter within the setup can be eliminated. This not only simplifies the system but also reduces costs and maintenance requirements. The affordability and ease of implementation make the model well-suited for widespread adoption in residential areas, contributing to improved water management practices and early detection of leaks.

In summary, the IoT-based water flow prediction model presented in this study offers a promising solution for optimizing water distribution and minimizing water wastage. Future research directions could focus on exploring the feasibility of the proposed solution for lower flow rates typically encountered in distribution pipelines. By continuing to enhance and refine such models, we can work towards achieving more efficient and sustainable water resource management systems.

Acknowledgement

References

[1] M. R. Islam, S. Azam, B. Shanmugam, and D. Mathur, "A review on current technologies and future direction of water leakage detection in water distribution network," *IEEE Access*, vol. 10, pp. 107 177–107 201, 2022.

[2] M. Islam and S. Aslan, "Leak detection and location pinpointing in water pipeline systems using a wireless sensor network," 05 2021, pp. 1–7.

[3] A. Lall, A. Khandelwal, R. Bose, N. Bawankar, N. Nilesh, A. Dwivedi, and S. Chaudhari, "Making analog water meter smart using ml and iot-based low-cost retrofitting," 08 2021, pp. 157–162.

[4] K. M. Chew, S. P. Yiiong, N. Bundan, and S. Tan, "The application of iot-based water pressure monitoring system," 09 2021, pp. 118–121.

[5] S. Sapre and J. P. Shinde, "Water pipeline monitoring on cloud leakage detection with a portable device," in *2019 IEEE Pune Section International Conference (PuneCon)*, 2019, pp. 1–5.

[6] A. Ayadi, O. Ghorbel, A. Obeid, M. Bensaleh, and M. Abid, "Leak detection in water pipeline by means of pressure measurements for wsn," 05 2017, pp. 1–6.

[7] *Pressure transmitter Type MBS 3000 and MBS 3050*, Danfoss, 2021. [Online]. Available: https://assets.danfoss.com/documents/latest/246349/AI244586497020en-001201.pdf