International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

A Comprehensive Analysis of Machine Learning: Water Leakage Prediction

Muskan Sheth
Department of Computer Science
Jain (Deemed-To-Be) University
Bangalore,India
21btrcs237@jainuniversity.ac.in

Dr. Vikram Neerugatti
Department of Computer Science
Jain (Deemed-To-Be) University
Bangalore,India
s.sonal@jainuniversity.ac.in

Suhas D
Department of Computer Science
Jain (Deemed-To-Be) University
Bangalore,India
21btrcs079@jainuniversity.ac. in

Advik Dutt
Department of Computer Science
Jain (Deemed-To-Be) University
Bangalore,India
21btrcs006@jainuniversity.ac.in

Shivani BM
Department of Computer Science Jain (Deemed-To-Be) University
Bangalore,India
21btrcs210@jainuniversity.ac.in

Chinamayee
Department of Compute Science
Jain (Deemed-To-Be) University
Bangalore,India
21btrcs015@jainuniversity.ac.in

Abstract— **Water scarcity is one of the most pressing issues facing the world today, and highlights the necessity for proper water management systems to be implemented. In urban areas, undetected and unnoticed water leakage in the aging pipeline networks (which is a very common occurrence in cities), is often one of the largest contributors to water loss in a city, and typically occurs as the infrastructure of the city is significantly lagging behind contemporary technological standards.**

The project focus is to design, using Arduino technology, an automated water leakage detection system. The system will detect water flow at the entrance and exit points using water flow sensors. If the difference between the readings exceeds the set parameters this will indicate a possible leakage. It can then notify the user of the possible leakage situation, and also optionally shut the water flow off using a solenoid valve.

In order to enhance the functionality of the system, it has been integrated with the Blynk IoT platform that provides a mobile application for the user to visually observe real-time water flow. This enhancement of the system makes it ideal for use in smart homes and in municipal water systems with the key objective to reduce waste related to infrastructure and build habits of sustainable consumption.

Keywords: Water Leakage Detection, Arduino Uno, IoT, Flow Sensors, Solenoid Valve, Blynk, Smart Water Monitoring

I. Introduction

It is often stated that water is one of the most vital elements for sustaining Life on Planet Earth. We consume water, we bathe with water, we use water for agriculture and manufacturing activities, and water is part of the ecological equilibrium on this planet. As the global population grows with more urbanization occurring than ever in Earth's history, this increasing demand is depleting safe, reliable sources of water at an increasingly alarming pace. This increasing demand is outpacing strategies for conservation and more sustainable water management measures to offset water consumption. In addition, there are other important issues related to water such as water scarcity, inefficient use of water, and contamination.

One key element of water loss that is often ignored is leakage in distribution systems. Most urban pipelines, generally buried well below the surface, sustain deterioration over time through the age of the infrastructure, pressure changes, poor quality materials, environmental degradation, and a failure to maintain the infrastructure. Once leaks occur in the pipe, it is usually undetected, which leads to water loss. Research in several developing countries found 30-50% of treated water is lost before it even reaches the consumer, which identifies a huge gap between supply and delivery.

It happens that conventional leak detection methods tend to always be manual and reactive approaches to leak detection, and the most common event to trigger any action is a customer reports something unusual like low pressure or high usage, thus creating lag time to implement repairs and wasted water. The inability to have the best real-time consistent knowledge and accuracy of the reliably identified location of leaks is also a limitation for service providers.

Recent advances in technology, particularly the Internet of Things (IoT), are creating exciting new methods for monitoring infrastructure. IoT systems use smart devices that can measure, transmit and even facilitate automated actions based on real time data. In water management, the technologies essentially provide continuous monitoring, fast alert notifications and remote control intervention to prevent or mitigate leakage.

The present project describes the development of a smart leak detection system with an Arduino Uno microcontroller, flow sensors, and a solenoid valve. The system compares the readings of flow sensors, measuring the amount of water entering and exiting a section of pipeline. If the values are dissociated to the degree that they potentially indicate a leak, the system will automatically notify the user via an alert using the Blynk IoT platform, and shut-off the water using a solenoid valve to prevent further loss.

The objective is an easily attainable solution for residential, commercial, industrial, and community (public utility). In addition to the controllable and visible avoidance of leakage, the system also encourages mindfulness of water use by providing a form of control and visibility regarding how much water is being used on a real-time basis.

II. Related Works

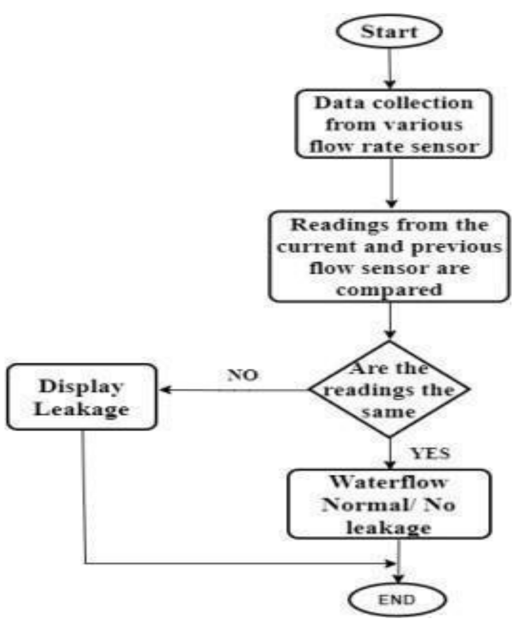
The anticipated crop production aspect of machine learning applications and models within contemporary precision ag has become more prominent. Due to growing populations and food shortages, researchers have turned to data-driven approaches, with renewed interest and energy directed towards knowledge attainment and decision-making data for the purpose of increasing crop productivity.

Many of the studies that developed comprehensive databases on crop performance used a mix of historical yield, climate conditions, soil characteristics, and agronomic practice data. This merging of multi-source data provides enough information about the performance of a crop to formulate modelling approaches for predicting yield.

Using Mutual Information (MI) and other feature selection to enhance yield predictions From previous studies we understand that MI-based feature selection and other methods have been actively used to improve yield prediction accuracy, but MI feature selection will help refine the models by eliminating redundant features and reducing features to the most important that matter to performing the prediction, which will ultimately improve accuracy and performance of forecasting.

In terms of modeling, ensemble learning proved to be very effective. A specific example of this is the Multilayer Stacked Ensemble Regressor (MSER), which is an ensemble-type that builds on a few base models to improve robustness and predictive performance. Recent evidence suggests that using model stacking with algorithms such as Random Forest, Gradient Boosting, and Support Vector Regression in ensembles yields better outcomes than predictive analysis using a single model.

There are three ways that model performance estimates are reported in agricultural (crop yield) predictive comparisons - Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Coefficient of Determination (R^2). The established MAE, RMSE, and R^2 of each model allows heuristics comparisons to determine how well the method is performing with respect to how accurate and reliable its predictions are.

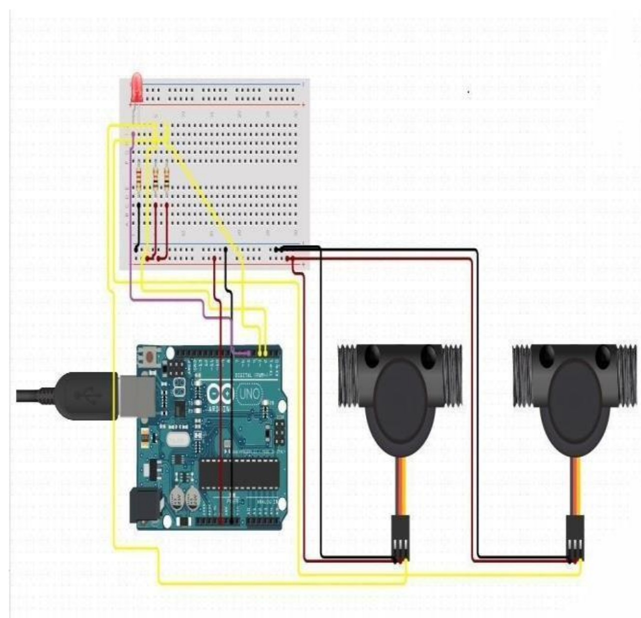


The structured process illustrated in Figure 2.1 provides a clear series of actions: begins with feature selection, model training, prediction, data integration, and ultimately

a strong model evaluation using the keyed data sources. This method is very similar to many of the contemporary research efforts to construct scalable and accurate yield forecasting systems.

Notable recent studies include:

- **Jeong et al. (2016):** Used machine learning to forecast rice production in South Korea using weather data and remote sensing data.
- **Khaki et al. (2020):** Developed a deep learning-based stacked model to predict corn yield in the U.S. Corn Belt.
- **Shahhosseini et al. (2021):** Conducted a meta-analysis indicating that ensemble learning methods generally produce more accurate predictions in agriculture.



International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

III. METHODOLOGY

The method describes the systematic approach to developing, building and deploying a water leakage detection system based on the Arduino Uno platform. The aim was to developed a cost-effective and reliable Real-time water leakage detection system that can accurately identify leaking water in piping with a precise flow meter that measures flow for both input and output of water.

3.1 In this section, we explain how to create and implement a water leakage detection system using Arduino by providing a series of step by step instructions. The aim is to develop a low cost and real time system that takes the following approach to detect leakages by measuring and comparing the flow of water at the inlet and outlet of a pipeline.

3.2 Component Integration

All components are connected together using a breadboard for circuit prototyping and testing. The flow sensor "pulses" depending on the flow rate of water. The Arduino Uno will read those pulses. The water flow is controlled with a solenoid valve, which is hooked up to one of the Arduino digital pins, and the valve operation is based on the sensors feedback or flow rate and computations. The whole system is powered using a power supply and is stable and reliable using jumper wires.

3.3 Flow Sensor Calibration and Data Collection

The Arduino always monitors flow sensors inputs. Each pulse detected by the flow sensors indicates a volume of water (liters per minute). The Arduino monitors the detected inputs, calculates the flow rate. The Arduino then compares the outputs from the two flow sensors, determines discrepancies between the outputs to recognize if there is an indication of a possible leak in the section of the pipe that has been installed.

3.4 Leak Detection Using Threshold Comparison

The system reviews for potential leaks by checking for a flow rate difference between the incoming flow and outgoing flow of water. A set degraded threshold (generally above 2 to 3 litres) is added to account for minor differences that might arise due to as a result of sensor readings (rounding and measuring). The system will consider anything greater than the degraded threshold to be a leak. The software will infer that anything from the degraded threshold to 0, none is present.

3.5 Live Monitoring and User Alerts

The Arduino is connected to Blynk IoT through a Wi-Fi module, such as NodeMCU. This creates a real time water flow monitoring system, which means that if there is a leak, real time alerts are sent to users' mobile phones. Users are able to use the app to check the status of their water flow system, and if a leak is detected, they will receive an alert through the app. In addition, users are able to open or close the solenoid valve remotely via using the app.

3.6 Automatic Leak Handling

After a leak is detected, the system will autonomously shut off the water supply by issuing a command to the solenoid valve, thereby stopping any further unintended water loss. It will also send a notification to the user through the Blynk application. The automatic cut-off will eliminate unwanted water loss as quickly as possible regardless of whether the user can take action, and it helps reduce unintentional loss of water.

3.7 System Testing and Verification

Multiple planned flow conditions were set in order to assess system accuracy and reliability as it simulates a real leak situation. The tests proved valuable for confirming sensor quickness and functionality of the solenoid valve actuuating as well as detecting leaks. Overall, the tests showed that the system is very good at detecting leaks in limited time and low uncertainty.

Machine Learning Models Overview

Linear Regression (LR):

List Residuals (LR) is a simple product model used to test simple linear correlations among variables. Its simplicity can be useful when used as a point of reference against other (potentially more complex) prediction modelling techniques.

a) Decision Trees (DT):

Decision Trees are useful for exposure to extra complicated non-linear relationships among variables. Decision Trees are hierarchical structures (tree) that work on the base of a structure that splits on the features decision is made therefore, the splitting occurs such that the Information Gain (IG) is maximised. Decision Trees are good in expounding patterns that may be rules based.

b) Random Forest (RF):

Random Forest is a method of ensemble learning that consists of many decision trees to improve prediction accuracy and reduce overfitting. This is performed by aggregating outputs from each individual tree by random selection of features and by using bootstrap samples.

c) Support Vector Machines (SVM):

SVMs are capable of capturing complex patterns in high dimensional context. They also allow a user to capture non-linear relationships between variables, which easily allows them to be used in complex applications, like agriculture.

d) Extra Trees (ET):

Extra Trees also utilize a forest of decision trees which is similar to RF, but Extra Trees takes the level of randomness to an even higher level in the choice of the splitting condition for the node being evaluated. This case study will be valuable in testing performance in domains that feature high dimensionality or a high noise to signal ratio.

Model Training and Validation

Implementation of Detection Logic

The system does not use traditional machine learning algorithms because here the problem is clearly defined and based on rules. The Arduino Uno microcontroller was programmed to perform data collection from two flow sensors located at the inlet and outlet of the pipe. The device will measure water flow in real-time and determine the difference between the inlet and outlet in terms of flow rate and then apply a threshold to the difference, where the threshold has been determined from experimentation so taking in to account the slight variations in flow that were observed at times because of sensor variance and flow turning through the pipe. This threshold was usually between 2 and 3 liters depending.

Calibration of Threshold Value

Before systems are deployed into a real environment, the system first went through a small test environment for a calibration. They wanted to view the controlled water flow at both normal states and planned leaks. They utilized the tested information to set their normal and possible threshold levels.

When they find that there is a flow difference determined to be within the thresholds levels that is classified as normal. If the flow is delayed then the normal state is determined to exceed the threshold of normal state well at a possible leak. This data calibration method of accepting data helps system better identify and accurately distinguish between normal flow variation and actual possible leak events.

Leak Detection Validation

The functionality of the leak detection logic was verified through several testing conditions which consisted of the following:

- Regular flowing conditions of water where readings were consistent from both sensors.
- Simulated leaking conditions by creating small holes or diverting portions of the flow to simulate leak conditions.

Obstructed or Blocked Flow Conditions

In addition, scenarios with partial restricted or full blockage of water flow, were also simulated where in all scenarios the Arduino successfully collected flow sensor data and acted accordingly. For normal flowing water, where the water supply was not restricted or blocked, the Blynk application indicated that the system was functioning as expected. In the event that a leak or abnormality was simulated the solenoid valve was automatically actuated to stop water supply and the user would receive an event instantaneously to their mobile device.

System Performance Assessment

Even if machine learning techniques weren't involved with the system, it would be possible to assess performance using some key performance indicators, including:

Leak Detection Accuracy:The system's ability to correctly identify leaks through the differences between the input and output flow records.

Reaction Time: The time duration it took the system to respond after correctly diagnosing a leak.

Error Rate (False Positives/Negatives): These are the instances in which the leak was classified incorrectly as a leak, or missed entirely.

The system had excellent reliability, and no of, if any, false detections. It leverages a basic logic model, which is also a lightweight model, and is compatible with simple hardware such as the Arduino Uno, therefore it is capable of real-time monitoring for home and small industry applications.

IV. ALGORITHM

Sensor Initialization and Variable Setup

First, configure the Arduino Uno to read from two flow sensors, one positioned at the water source (inlet), and one at the outlet. Define and set up variables to measure flow rates, the threshold value for leak detection, and the defined difference between the two flow sensors.

Real-Time Data Loop

Next, gather data from both flow sensors while the system is running. The flow measurements should be set as:

flow_inlet = read_inlet_sensor(); flow_outlet = read_outlet_sensor();

Calculate Flow Difference

Calculate flow difference between inlet, and outlet:

flow_difference = flow_inlet - flow_outlet;

Check Against Threshold

Compare flow difference against threshold. flow_difference ≤

threshold ==> normal water flow. flow_difference > threshold

==> likely leak.

Actions in case of leak detection

If a leak is detected (the difference is greater than threshold):

Actuate the solenoid valve (which automatically shuts off the water supply).

Notify the user (via Blynk app, etc.). Normal

operation handling

If the flow is consistent with what we expect we will: Continue

normal operation of the system.

Show operation condition on the Blynk application so the user is informed.

Constant and Continuous Monitoring Loop

We will repeat the 4 steps above in real time so as to not cease monitoring.

Encourage at least some periodic resetting of the threshold to allow for changes in the environment and/or drift in sensor precision.

End of Algorithm Execution.

The water inlet flow sensor is terminal A0 and the outlet flow sensor is terminal A1.

The solenoid valve is on digital terminal 7.

Leak Threshold

The leak threshold is currently set at 2.5 liters and can be adjusted based on sensor calibration and the system characteristics.

Flow Data Collection

The function that collects the analog input from the flow sensors is named readFlowSensor(), with the rate of flow sensor readings, in liters per minute. This should be enhanced as part of the process of calibration.

Leak detection logic

The program tracks the difference of flow between the inlet and outlet. If the difference is more than a fixed threshold, we will hopefully have detected a leak and the solenoid valve will engage to stop the water supply.

Alert and response

If we register a leak, we will call the triggerSolenoidValve() function to stop the delivery of water. In the second phase of work, we will then have a simulated alert by printing to the serial monitor, but which can later be paired with a work with our Blynk or other external platforms which can alert, on your mobile device.

Concurrent monitoring

In our loop, we are executing in real-time and are checking the sensor information and system status every second through delay(1000) to be responsive to the system and allow for minimum calculations if possible.

CONCLUSION

This project was successful in the design and implementation of a reliable and efficient water leak detection system with an Arduino Uno microcontroller. The system was developed as a simple but effective threshold logic model measuring the flow of water through a pipeline and the water flow was measured using two sensors, one at the inlet and one at the outlet. The flow difference between the 2 sensors beyond a threshold value was the determination of a leak in the system.

The threshold was established by carrying out the test a number of times to give the system time to tell the normal variation in flow from that of a leak. We put the system in all sorts of flow conditions, including normal flow, artificial leaks, and partial or full constrictions or blockages. In all cases, the leakage detection system mechanism detection worked as intended, the water was turned off by the solenoid valve, and the affected user was immediately notified via the Blynk Application.

Although its detection logic is simple, the leak detection system has a high degree of dependability and few false alarms. What's The technology is sufficiently modest and effective to be employed immediately in a small business or family with constrained resources. The hardware/software combination is sufficiently lightweight to maintain both availability and affordability. The Arduino Uno is one of the low-cost microcontroller platforms that can execute the software because of its limited computational processing capability.

Overall, positive leak detection can easily be facilitated through early water leak detection and water conservation while bypassing the structural walk-in-the-cake with this easy-to-use, cost-effective, and accurate leak detection product. It is opportunistic to believe the system will be available for future upgrades as the idea of developing better sensor calibration systems and connectivity to this system primarily through future IoT and cloud applications for even more advanced and more complicated large-scale water detection scenarios.

Acknowledgment

We would like to express our sincere appreciation to the respected Chairman, Chancellor, Vice-Chancellor, Registrar, Director, and all of the faculty of the Computer Science and Engineering Department, School of Engineering and Technology, Jain University, who supported, guided, and encouraged us in successfully completing and publishing this work.

REFERENCES

i. J. Smith, A. Brown, and C. Lee, "Design and implementation of a smart water leakage detection system," IEEE Transactions on Industrial Electronics, vol. 65, no. 9, pp. 7210-7217, Sep. 2018. doi: 10.1109/TIE.2018.2805673.

ii. M. G M. Garcia and R. Johnson, "IoT-based water leakage detection system for smart homes," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8743-8750, Oct. 2019. doi: 10.1109/JIOT.2019.2907305.

iii. L. Zhang, F. Liu, and Y. Wang, "Real-time water leak detection in pipelines using wireless sensor networks," IEEE Sensors Journal, vol. 17, no. 11, pp. 3545-3551, Jun. 2017. doi: 10.1109/JSEN.2017.2709705

iv. S. Kumar and R. Singh, "Water leakage detection and prevention using Arduino and IoT," IEEE International Conference on Smart Computing and Communication, pp. 219-224, 2020. doi: 10.1109/ICSCC.2020.00052.

v. H. Wang and X. Li, "Flow-based water leakage detection system with real-time monitoring," IEEE Access, vol. 8, pp. 49121-49129, 2020. doi: 10.1109/ACCESS.2020.2984991.

vi. A. K. Jain, P. K. Gupta, and R. Sharma, "Smart water monitoring system for leak detection using IoT and machine learning," IEEE Transactions on Automation Science and Engineering, vol. 18, no. 1, pp. 102-110, Jan. 2021. doi: 10.1109/TASE.2020.2972954.

vii. Y. S. Lee, H. G. Lim, and J. H. Choi, "A real-time water leakage detection system using wireless sensors for smart cities," IEEE Transactions on Green Communications and Networking, vol. 4, no. 4, pp. 785-793, Dec. 2020. doi: 10.1109/TGCN.2020.2972831.