



IoT Based Smart Fluid Flow Meter

C.Ramuvel, Srithar S, Subasri J, Sundharavel A

Assistant Professor, Dept. of ECE, MAM School of Engineering, Siruganur, Trichy, Tamil Nadu, India

Student, MAM School of Engineering, Siruganur, Trichy, Tamil Nadu, India

Student, MAM School of Engineering, Siruganur, Trichy, Tamil Nadu, India

Student, MAM School of Engineering, Siruganur, Trichy, Tamil Nadu, India

Publication History: Received: 25.02.2026; Revised: 20.03.2026; Accepted: 25.03.2026; Published: 28.03.2026.

ABSTRACT: Efficient management and monitoring of fluid systems are essential in many sectors such as industrial processes, residential water supply, and agricultural irrigation. With increasing demand for water and other fluid resources, it is important to develop intelligent systems that can monitor fluid consumption accurately and prevent unnecessary wastage. Traditional flow monitoring methods often require manual inspection and lack real-time data analysis capabilities, which can lead to inefficient resource utilization and delayed detection of leaks or abnormal flow conditions. To address these challenges, this project proposes an IoT-based Smart Flow Meter that enables continuous monitoring of fluid flow in pipelines and provides real-time data access through internet connectivity. The proposed system uses a flow sensor to measure the volume and rate of fluid passing through a pipeline. The sensor generates pulses proportional to the flow rate, which are then read and processed by an Arduino UNO microcontroller. The microcontroller calculates the flow rate and total volume of fluid passing through the pipe. This data is then transmitted to a cloud server using Internet of Things (IoT) technology. The integration of IoT allows users to monitor fluid usage remotely through web-based or mobile applications. This remote monitoring capability provides convenience and improves system efficiency by allowing users to access real-time data from anywhere.

KEYWORDS: IoT- Smart Fluid Flow Meter, Flow Measurement, Automation- Real-time Monitoring, Sensor Technology, Industrial IoT, Water Management

I. INTRODUCTION

1.1 Overview

Water and fluid management has become an important concern in modern society. With the rapid growth of industries, urban areas, and agricultural activities, the demand for water and other fluid resources has increased significantly. Efficient monitoring and management of fluid systems are essential to ensure proper utilization of resources and to prevent wastage. Traditional methods of monitoring fluid flow in pipelines often rely on manual readings and mechanical meters, which may not provide real-time information and may lead to inaccurate data. In recent years, technological advancements in embedded systems and Internet of Things (IoT) have enabled the development of intelligent monitoring systems. These systems can automatically measure, analyze, and transmit data over the internet, allowing users to monitor systems remotely. One such application is the IoT-based smart flow meter, which is designed to measure and monitor the flow of fluids in pipelines while providing real-time information to users. The proposed IoT-based smart flow meter system uses a flow sensor to measure the flow rate and volume of fluid passing through a pipeline. The data collected by the sensor is processed using a microcontroller such as Arduino UNO. The processed data is then transmitted to a cloud server through an IoT communication module. Users can access the data through web applications or mobile devices, enabling remote monitoring and control. This project aims to develop a reliable and efficient system that helps users monitor fluid usage in real time, detect leaks, and manage resources effectively. By integrating sensor technology with IoT communication, the smart flow meter provides an intelligent solution for fluid monitoring and management.

II. PROBLEM STATEMENT

Traditional flow monitoring systems often rely on manual measurements and mechanical meters that do not provide real-time data or remote monitoring capabilities. These limitations can lead to inefficient resource management, delayed leak detection, and increased operational costs.



In many water distribution systems, leaks and excessive consumption remain undetected for long periods, resulting in significant resource wastage. There is a need for an intelligent monitoring system that can continuously measure fluid flow, transmit data remotely, and provide alerts in case of abnormal conditions.

The proposed IoT-based smart flow meter addresses these challenges by integrating sensor technology with IoT communication to enable real-time monitoring and efficient resource management.

III. EXISTING SYSTEM

In traditional fluid management systems, the measurement of fluid flow is commonly performed using conventional mechanical flow meters. These meters measure the volume of fluid passing through a pipeline using mechanical components such as turbines or rotating blades. The readings are usually displayed locally on the meter and require manual observation and recording by operators. In many residential and agricultural applications, water usage is monitored only periodically, which makes it difficult to track real-time consumption.

One of the major limitations of the existing system is the lack of continuous monitoring. Since the system depends on manual reading, it is difficult to detect sudden changes in flow rate or leakage in pipelines. If a pipe breaks or leaks, the problem may remain unnoticed for a long time, leading to significant wastage of water or other fluids. Additionally, traditional flow meters do not provide remote access to data, making it impossible for users to monitor the system from distant locations.

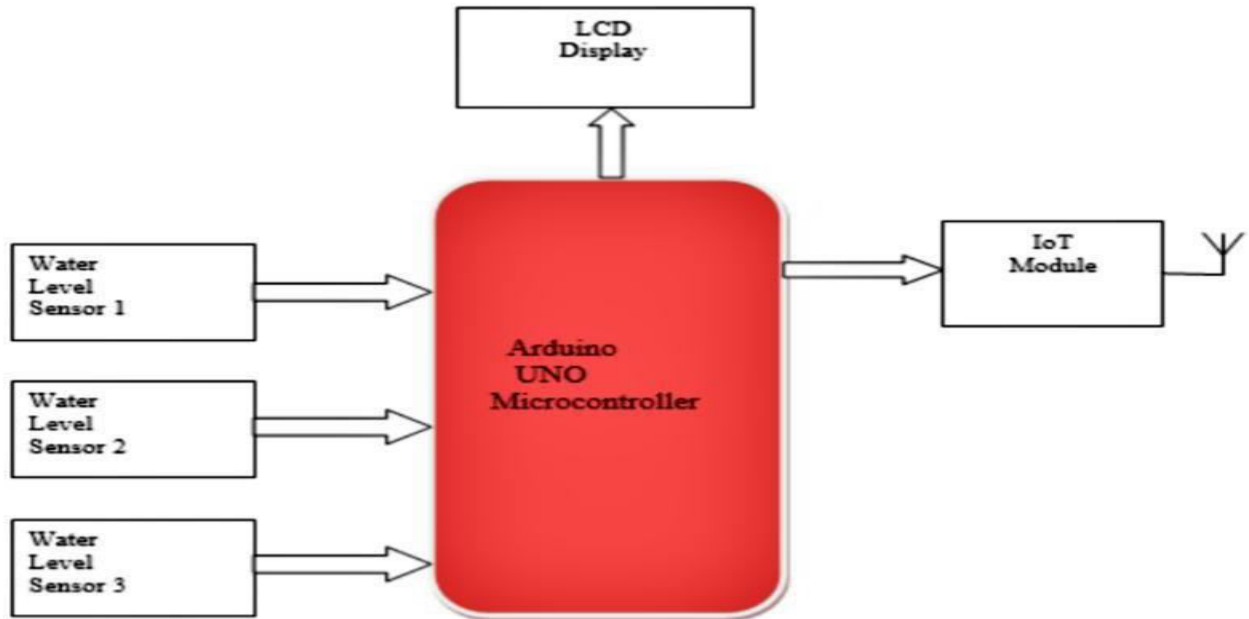
IV. PROPOSED SYSTEM

To overcome the limitations of traditional systems, the proposed project introduces an IoT-based Smart Flow Meter for real-time monitoring and efficient management of fluid flow in pipelines. The system uses a flow sensor to measure the flow rate and total volume of fluid passing through the pipe. The sensor outputs pulses corresponding to the flow of liquid, which are processed by a microcontroller such as an Arduino UNO. The Arduino UNO collects the sensor data and calculates the flow rate and total fluid consumption. The processed data is then transmitted to a cloud server through IoT connectivity.

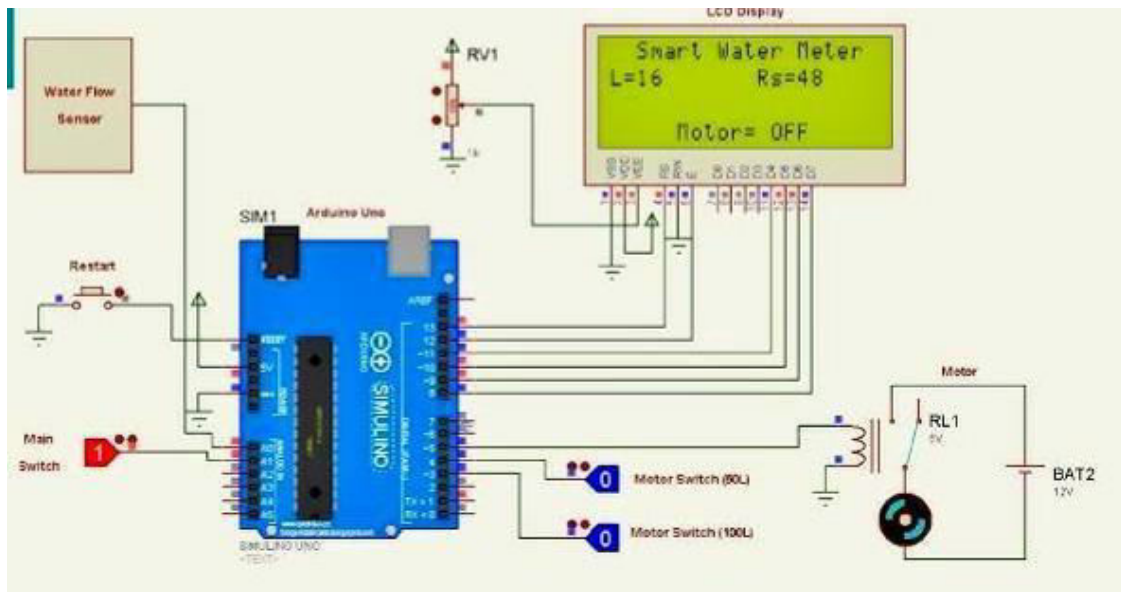
This enables users to monitor the system remotely using web or mobile applications. Real-time data visualization allows users to observe fluid usage patterns and make better decisions regarding resource management.

One of the important features of the proposed system is automatic leak detection. If the system detects abnormal flow conditions or unusual changes in flow rate, it can generate alerts and notifications for the user. This helps in identifying problems quickly and taking immediate corrective action to prevent wastage. The system also supports data storage and analysis. Historical data collected from the flow sensor can be stored on the cloud platform, allowing users to analyze usage trends over time. This feature is useful for improving efficiency in industrial, agricultural, and residential applications.

V. BLOCK DIAGRAM



BLOCK DIAGRAM



VI. DESIGN AND SELECTION OF COMPONENTS

A. Arduino Uno

Arduino is the main part of the project; hence it should be selected properly. In our project we have selected Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter. Arduino Uno has a number of facilities for communicating with a computer, another Arduino board, or other microcontrollers.



B. WATER FLOW SENSIR

Here the rotating wheel is fixed in the inside pipe where the waterflow has to measure. When the water is flowing with pressure, the wheel is rotating. The wheel rotation is monitored by the proximity sensor. Rotating wheel is nothing but a Plus shape magnet. The proximity sensor is delivered the output in the form of pulse when it detects the magnetic flux. This is given to microcontroller. The microcontroller counts the pulse which is equal to rate of waterflow in the pipe. This circuit is mainly used to monitoring the water flow and maintains the set level using control circuit.

C. IOT MODULE

The Wi-Fi library for ESP8266 has been developed based on ESP8266 SDK, using the naming conventions and overall functionality philosophy of the Arduino WiFi library. Over time, the wealth of Wi-Fi features ported from ESP9266 SDK to esp8266 / Arduino outgrew Arduino WiFi library and it became apparent that we would need to provide separate documentation on what is new and extra. This documentation will walk you through several classes, methods and properties of the ESP8266WiFi library. If you are new to C++ and Arduino, don't worry. We will start from general concepts and then move to detailed description of members of each particular class including usage examples.

D. POWER

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (nonUSB) power can come either from an AC to- DC adapter (wall -wart) or battery. The adapter can be connected by plugging a 2.1mm center -positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

E. Soft Access Point

An access point (AP) is a device that provides access to a Wi-Fi network to other devices (stations) and connects them to a wired network. The ESP8266 can provide similar functionality, except it does not have interface to a wired network. Such mode of operation is called soft access point (soft-AP). The maximum number of stations that can simultaneously be connected to the soft-AP can be set from 0 to 8, but defaults to 4.

The soft-AP mode is often used as an intermediate step before connecting ESP to a Wi-Fi in a station mode. This is when SSID and password to such network is not known upfront. ESP first boots in soft-AP mode, so we can connect to it using a laptop or a mobile phone. Then we are able to provide credentials to the target network. Then, the ESP is switched to the station mode and can connect to the target Wi-Fi.

F. LCD Display / LED Indicators

These components are used to provide user feedback.

Function:

- Displays detected fabric type
- Shows temperature status
- Indicates system operation (ON/OFF, heating, etc.)

VII. SOFTWARE INSTALLATION

Arduino IDE and embedded c program are used as software tools.

A. ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is an open-source software platform used to develop and upload programs to Arduino boards and other compatible microcontrollers. It provides a user-friendly interface that simplifies the process of writing and testing embedded system programs. Arduino IDE supports multiple operating systems including Windows, Linux, and macOS, making it accessible to a wide range of users. The environment contains a built-in text editor where programmers can write their code, known as a sketch. These sketches are written using a simplified version of the C/C++ programming language.

The Arduino IDE includes several important components that help developers during program development. The code editor is used to write and modify the program. The toolbar provides quick access to commonly used

functions such as verifying the program, uploading code to the microcontroller, and saving files. The message area and console display error messages, compilation results, and debugging information. One of the major advantages of Arduino IDE is the availability of a large collection of libraries. These libraries simplify programming by providing ready-made functions for sensors, communication modules, displays, and other hardware devices. By using these libraries, developers can easily integrate different components into their projects without writing complex code.

Another important feature of Arduino IDE is the Serial Monitor, which allows developers to view real-time data from the microcontroller. The binary code of Arduino is illustrated in Figure A

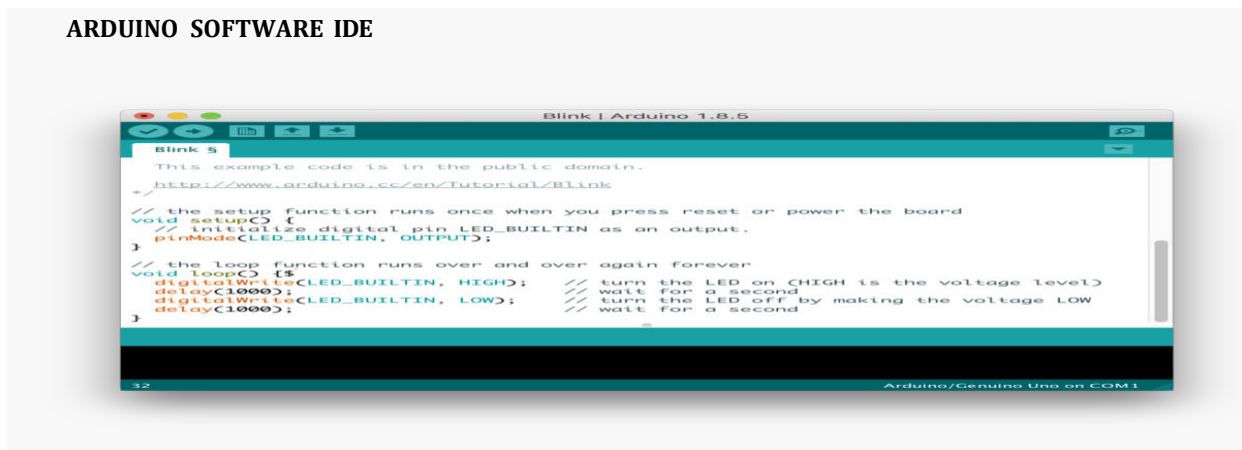


Fig A. Binary code of Arduino

C. Embedded C

Embedded C is a programming language used for developing software in embedded systems. It is an extension of the C programming language and is specifically designed for programming microcontrollers and hardware-based devices.

Embedded C provides direct access to hardware components such as input/output ports, timers, communication modules, and memory registers. This allows programmers to control sensors, actuators, and other electronic devices efficiently. In embedded system development, the program must interact closely with the hardware. Embedded C allows the programmer to read data from sensors, perform logical operations, and control output devices such as LEDs, motors, buzzers, and displays.

A typical Embedded C program contains several sections, including header files, variable declarations, initialization routines, and the main program loop. In Arduino-based systems, the program structure mainly consists of two functions: `setup()` and `loop()`. The `setup()` function is executed once when the microcontroller starts running. It is used to initialize input and output pins, configure communication modules, and prepare sensors for operation. The `loop()` function runs continuously after the `setup` function. It performs the main tasks of the system such as reading sensor values, processing data, and controlling output devices. Since the loop runs repeatedly, the system can continuously monitor inputs and respond to changes in real time.

Embedded C supports various programming structures such as conditional statements, loops, and functions. Conditional statements such as `if-else` allow the program to make decisions based on sensor readings. Loops such as `for` and `while` are used for repetitive operations.

Embedded C also supports communication protocols such as UART, SPI, and I2C, which allow the microcontroller to communicate with external devices. These protocols enable data exchange between sensors, displays, memory devices, and IoT communication modules. Because embedded systems often operate in real-time environments, Embedded C programs must be optimized for speed, memory efficiency, and reliability. Proper programming ensures that the system



responds quickly and operates without errors. Embedded software refers to computer software designed to control non-computer devices, known as embedded systems. It is optimized for the specific hardware it runs on and is subject to constraints such as time and memory limitations. Embedded software is often referred to interchangeably with firmware.

A defining characteristic of embedded software is that it may not be directly initiated or controlled via a human interface, relying instead on machine interfaces. Manufacturers integrate embedded software into various devices, including cars, telephones, modems, robots, appliances, toys, security systems, pacemakers, televisions, set-top boxes, and digital watches. The complexity of embedded software can vary greatly, from simple lighting controls on a microcontroller with limited memory to sophisticated applications requiring advanced computation frameworks.

VIII. RESULT AND ANALYSIS

The IoT-based Smart Flow Meter system was successfully designed and implemented using a flow sensor, Arduino UNO microcontroller, and IoT connectivity for real-time monitoring. The developed system was tested under different flow conditions to evaluate its performance, accuracy, and reliability. The flow sensor was able to measure the fluid passing through the pipeline and generate pulse signals proportional to the flow rate. These signals were processed by the Arduino UNO to calculate the flow rate and total volume of fluid. During the testing phase, the system demonstrated accurate measurement of fluid flow with minimal error.

IX. CONCLUSION

The IoT-based Smart Flow Meter project presents an efficient and intelligent solution for monitoring fluid flow in pipelines. The system integrates a flow sensor, Arduino UNO microcontroller, and IoT technology to measure fluid flow and transmit data for real-time monitoring. By using IoT connectivity, users can access flow information remotely through web or mobile applications, which improves convenience and system management.

The developed system helps in detecting leaks, monitoring fluid consumption, and reducing unnecessary wastage of water or other fluids. The ability to store and analyze historical data further enhances the effectiveness of the system in optimizing resource usage. The use of low-cost components such as Arduino and flow sensors makes the system affordable and easy to implement in different environments.

REFERENCES

1. R. K. Sharma and S. Verma, "IoT-Based Smart Water Flow Monitoring System," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 4, pp. 1–7, 2020.
2. P. S. Reddy and M. K. Singh, "Smart Water Metering Using IoT and Arduino," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 8, no. 2, pp. 1–6, 2019.
3. Kumar and S. Joshi, "Real-Time Monitoring of Fluid Flow Using IoT," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 10, no. 5, pp. 1–8, 2021.
4. C.Nagarajan and M.Madheswaran - 'Stability Analysis of Series Parallel Resonant Converter with Fuzzy Logic Controller Using State Space Techniques'- Taylor & Francis, *Electric Power Components and Systems*, Vol.39 (8), pp.780-793, May 2011. DOI: 10.1080/15325008.2010.541746
5. C.Nagarajan and M.Madheswaran - 'Experimental verification and stability state space analysis of CLL-T Series Parallel Resonant Converter' - *Journal of Electrical Engineering*, Vol.63 (6), pp.365-372, Dec.2012. DOI: 10.2478/v10187-012-0054-2
6. C.Nagarajan and M.Madheswaran - 'Performance Analysis of LCL-T Resonant Converter with Fuzzy/PID Using State Space Analysis'- Springer, *Electrical Engineering*, Vol.93 (3), pp.167-178, September 2011. DOI 10.1007/s00202-011-0203-9
7. S.Tamilselvi, R.Prakash, C.Nagarajan, "Solar System Integrated Smart Grid Utilizing Hybrid Coot-Genetic Algorithm Optimized ANN Controller" *Iranian Journal Of Science And Technology-Transactions Of Electrical Engineering*, DOI10.1007/s40998-025-00917-z,2025
8. S.Tamilselvi, R.Prakash, C.Nagarajan, " Adaptive sliding mode control of multilevel grid-connected inverters using reinforcement learning for enhanced LVRT performance" *Electric Power Systems Research* 253 (2026) 112428, doi.org/10.1016/j.epr.2025.112428



9. S.Thirunavukkarasu, C. Nagarajan, 2024, "Performance Investigation on OCF and SCF study in BLDC machine using FTANN Controller," *Journal of Electrical Engineering And Technology*, Volume 20, pages 2675–2688, (2025), doi.org/10.1007/s42835-024-02126-w
10. Nagarajan, M.Madheswaran and D.Ramasubramanian- 'Development of DSP based Robust Control Method for General Resonant Converter Topologies using Transfer Function Model'- *Acta Electrotechnica et Informatica Journal* , Vol.13 (2), pp.18-31, April-June.2013, DOI: 10.2478/aei-2013-0025.
11. C.Nagarajan and M.Madheswaran - 'DSP Based Fuzzy Controller for Series Parallel Resonant converter'- Springer, *Frontiers of Electrical and Electronic Engineering*, Vol. 7(4), pp. 438-446, Dec.12. DOI 10.1007/s11460-012-0212-0.
12. C.Nagarajan and M.Madheswaran - 'Experimental Study and steady state stability analysis of CLL-T Series Parallel Resonant Converter with Fuzzy controller using State Space Analysis'- *Iranian Journal of Electrical & Electronic Engineering*, Vol.8 (3), pp.259-267, September 2012.
13. C.Nagarajan and M.Madheswaran, "Analysis and Simulation of LCL Series Resonant Full Bridge Converter Using PWM Technique with Load Independent Operation" has been presented in ICTES'08, a IEEE / IET International Conference organized by M.G.R.University, Chennai.Vol.no.1, pp.190-195, Dec.2007
14. Suganthi Mullainathan, Ramesh Natarajan, "An SPSS and CNN modelling based quality assessment using ceramic materials and membrane filtration techniques", *Revista Materia (Rio J.)* Vol. 30, 2025, DOI: <https://doi.org/10.1590/1517-7076-RMAT-2024-0721>
15. M Suganthi, N Ramesh, "Treatment of water using natural zeolite as membrane filter", *Journal of Environmental Protection and Ecology*, Volume 23, Issue 2, pp: 520-530,2022
16. V. Patel and R. Desai, "IoT-Based Industrial Fluid Flow Monitoring System," *International Journal of Computer Applications*, vol. 178, no. 45, pp. 1–7, 2020.
17. Applications, vol. 178, no. 45, pp. 1–7, 2020.
18. M. R. Khan and F. Ahmed, "Wireless Smart Flow Meter for Water Distribution Systems," *International Journal of Computer Applications*, vol. 7, no. 3, pp. 1–6, 2019.
19. *Electrical and Electronics Research*, vol. 7, no. 3, pp. 1–6, 2019.
20. **Kiran, A., Rubini, P., & Kumar, S. S. (2025)**. Comprehensive review of privacy, utility and fairness offered by synthetic data. *IEEE Access*.
21. **Gopinathan, V. R. (2024)**. Real-Time Financial Risk Intelligence Using Secure-by-Design AI in SAP-Enabled Cloud Digital Banking. *International Journal of Computer Technology and Electronics Communication*, 7(6), 9837-9845.
22. **Udayakumar, R., Elankavi, R., Vimal, R., & Sugumar, R. (2023)**. Improved Particle Swarm Optimization with Deep Learning-Based Municipal Solid Waste Management in Smart Cities. *Environmental & Social Management Journal*, 17(4).
23. **Anand, L. (2023)**. An Intelligent AI and ML-Driven Cloud Security Framework for Financial Workflows and Wastewater Analytics. *International Journal of Humanities and Information Technology*, 5(02), 87-94.
24. **Soundappan, S. J. (2020)**. Big Data Analytics in Healthcare: Applications for Pandemic Forecasting. *International Journal of Advanced Research in Computer Science & Technology*, 3(1), 2248-2253.
25. **Rajasekar, M. (2024)**. Real-Time Predictive DevOps Intelligence for Risk-Aware Digital Business Processes in Cloud and SAP Ecosystems. *International Journal of Advanced Research in Computer Science & Technology*, 7(4), 10713-10718.
26. **Poornima, G., & Anand, L. (2024, May)**. Novel AI Multimodal Approach for Combating Against Pulmonary Carcinoma. In *2024 5th International Conference for Emerging Technology (INCET)* (pp. 1-6). IEEE.
27. **Prabha, P. S., & Rengarajan, A. (2025)**. Adaptive Cloud Resource Allocation Using Attention-Driven Deep Reinforcement Learning. *Engineering, Technology & Applied Science Research*, 15(6), 29334-29340.
28. **Jagadeesh, S., & Sugumar, R. (2017)**. A Comparative study on Artificial Bee Colony with modified ABC algorithm. *European Journal of Applied Sciences*, 9(5), 243-248.
29. **Varma, K. K., & Anand, L. (2025, March)**. Deep Learning Driven Proactive Auto Scaler for High-Quality Cloud Services. In *International Conference on Computing and Communication Systems for Industrial Applications* (pp. 329-338). Singapore: Springer Nature Singapore.
30. **Kumar, S. A., & Anand, L. (2025)**. A Novel EEG-Based Deep Learning Framework for Enhancing Communication in Locked-In Syndrome Using P300 Speller and Attention Mechanisms. *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS*, 19(11), 3841-3855.
31. **Poornima, G., & Anand, L. (2025)**. Medical image fusion model using CT and MRI images based on dual scale weighted fusion based residual attention network with encoder-decoder architecture. *Biomedical Signal Processing and Control*, 108, 107932.



32. Archana, R., & Anand, L. (2025). Residual u-net with Self-Attention based deep convolutional adaptive capsule network for liver cancer segmentation and classification. *Biomedical Signal Processing and Control*, 105, 107665. Kumar, S. A., & Anand, L. (2025). A Novel EEG-Based Deep Learning Framework for Enhancing Communication in Locked-In Syndrome Using P300 Speller and Attention Mechanisms. *KSII Transactions on Internet and Information Systems*, 19(11), 3841-3855.
33. Rengarajan, A. (2025). Cloud-Based AI-Driven Threat Detection Framework for Smart Grid Cybersecurity. *International Journal of Future Innovative Science and Technology*, 8(6), 16065.
34. Murugeswari, B., Sudharson, K., Panimalar, S. P., Shanmugapriya, M., & Abinaya, M. (2020). SAFE-Secure Authentication in Federated Environment using CEG Key code.
35. Raj A. A., & Sugumar, R. (2023). Early Detection of COVID-19 with Impact on Cardiovascular Complications using CNN Utilising Pre-Processed Chest X-Ray Images. 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), IEEE.
36. Jagadeesh, S., & Sugumar, R. (2017). A Comparative study on Artificial Bee Colony with modified ABC algorithm. *European Journal of Applied Sciences*, 9(5), 243-248.
37. Selvi, G. V., Anbarasan, A. B., Murthy, B. A., & Prabavathy, S. (2023). An Application Oriented Integrated Unequal Clustering Algorithm for Wireless Sensor Network. In *Underwater Vehicle Control and Communication Systems Based on Machine Learning Techniques* (pp. 140-154). CRC Press.
38. Sruthi, R. S., Ananya, S., & Murugeswari, B. (2010). Web Based Virtual Control System Laboratory and On-Line Temperature Control of Electrophoresis Equipment using LabVIEW. *International Journal of Computer Applications*, 975, 8887.
39. Vimal Raja, G. (2021). Mining Customer Sentiments from Financial Feedback and Reviews using Data Mining Algorithms. *International Journal of Innovative Research in Computer and Communication Engineering*, 9(12), 14705-14710.
40. MATHEW, A. R. (2025). Neurosecurity and Brain-Computer Interfaces.
41. Soundappan, S. J. (2024). AI-Driven Customer Intelligence in Enterprise Lakehouse Systems Sentiment Mining Governance-Aware Analytics and Real-Time Data Synchronization. *International Journal of Advanced Engineering Science and Information Technology (IJAESIT)*, 7(5), 14905.
42. Mathew, A. (2025). Human-AI Collaboration in Security Operations: Measuring Alert Trust, Automation Bias, and Analyst Upskilling in AI-Augmented SOC Environments. *International Journal of Computer Technology and Electronics Communication*, 8(5), 11375-11380.
43. Soundappan, S. J. (2022). AI-Based Fault Detection and Isolation for Reliability in Modern Power Systems. *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)*, 5(4), 7106-7110.
44. Poornima, G., & Anand, L. (2024, April). Effective Machine Learning Methods for the Detection of Pulmonary Carcinoma. In *2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)* (pp. 1-7). IEEE. Garg, V. K., Soundappan, S. J., & Kaur, E. M. (2020). Enhancement in intrusion detection system for WLAN using genetic algorithms. *South Asian Research Journal of Engineering and Technology*, 2(6), 62-64.
45. Rengarajan, A., Jayakumar, C., & Sugumar, R. (2012). Optimization Of Recent Attacks Using Internet Protocol. *National Journal of System and Information Technology*, 5(1), 8.
46. Mathew, A. (2024). AI TRiSM: Trust, Risk, and Security Management in Cybersecurity. *Cybersecurity*, 4(3), 84-90.
47. Mathew, A. (2025). Deep seek vs. ChatGPT: A deep dive into AI Language mastery. *Int J Multidisciplinary Res*, 7(1), 1-5.