Journal on Electronic and Automation Engineering

Vol: 4(2), June 2025

REST Publisher; ISSN: 2583-6951 (Online)

Website: https://restpublisher.com/journals/jeae/

DOI: https://doi.org/10.46632/jeae/4/2/56

IoT Based Smart Border Security System

***G.D. Sivajagan, P. Gayatri, P. Urjitha, R. Ramya, V. Sai Sravanthi**

Nalla Malla Reddy Engineering College, Hyderabad, Telangana, India.

*Corresponding Author Email: shivajagan.ece@nmrec.edu.in

Abstract: *This paper presents a comprehensive, cost-effective, and intelligent surveillance system utilizing the AI Thinker ESP32-CAM module, integrating real-time video streaming, human detection, and audible alerts for enhanced security. Designed for applications such as home monitoring, industrial safety, and especially border surveillance, the system leverages a PIR motion sensor to trigger video capture and streaming via a local network. With video frames processed through a locally hosted Flask server using YOLO and Open CV for object detection, the presence of a person activates a buzzer and enables instant alerts through IoT connectivity. By combining low-cost hardware, open-source software, and deep learning techniques, including convolutional neural networks for robust performance in low-light and complex scenarios, the project offers a scalable and autonomous security solution. Specifically addressing India's extensive 15,106 km border with its neighboring countries, this system demonstrates significant potential for securing restricted and high-risk areas, effectively merging AI and IoT to meet modern surveillance demands.*

1. INTRODUCTION

Advancements in AI, IoT, and embedded systems have led to the development of intelligent surveillance solutions to address the limitations of traditional security systems like CCTV. The ESP32-CAM module, with its camera, Wi-Fi connectivity, and GPIOs, is used in a real-time human detection system. This system utilizes a PIR motion sensor and a YOLO deep learning model to detect humans and trigger alerts via a buzzer. The system provides a smart, cost-effective alternative for applications in smart homes, infrastructure, and border security. In border security, it offers an innovative IoT-based solution for detecting intruders using deep learning techniques and real-time notifications, enhancing surveillance in remote areas and challenging conditions.

2. LITERATURE SURVEY

The current project combines IoT, image processing, and sensor technologies to create a robust and scalable smart surveillance system. Several research efforts have contributed to this field, such as Siham Boukhalfa's IoT-based border security system using drones for monitoring, although it faces challenges in low-light conditions. M. Poojitha Yadav's radar vision system combines ultrasonic and infrared sensors for accurate motion detection in border defense. Vidhya Lavanya's system uses thermal imaging cameras (FLIR sensors) for detecting intrusions in low-visibility environments. Anooja Ali's project employs a Raspberry Pi and NodeMCU with infrared sensors for efficient motion detection and communication, enhancing smart surveillance and automation.

3. OBJECTIVE OF PROJECT

This project aims to develop a smart surveillance system using the ESP32-CAM module for real-time video streaming and human detection. By integrating the YOLO object detection algorithm with Open CV on a local Flask server, the system accurately identifies human presence and triggers an alarm. The goal is to create a low-cost, scalable, and autonomous security solution that combines IoT, edge computing, and open-source technologies, providing efficient operation in remote or challenging environments with minimal network dependency.

4. BLOCK DIAGRAM AND ITS DESCRIPTION

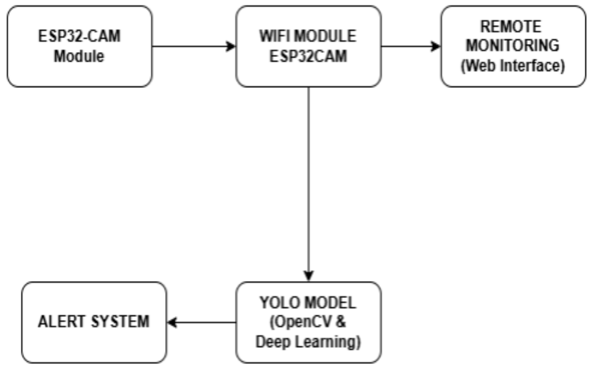

FIGURE 1. Block Diagram of IOT based Smart Border Security System

Security System:
The smart surveillance system uses a PIR sensor to detect motion and trigger the ESP32-CAM module, which captures images or streams video. The captured data is sent to a server via Wi-Fi for processing. A Python Flask server running the YOLO deep learning model with Open CV detects human presence in the media. If a person is detected, the system activates a buzzer for a local alert and can send real-time notifications, such as emails or app alerts, while storing logs and images in the cloud.

Flowchart:

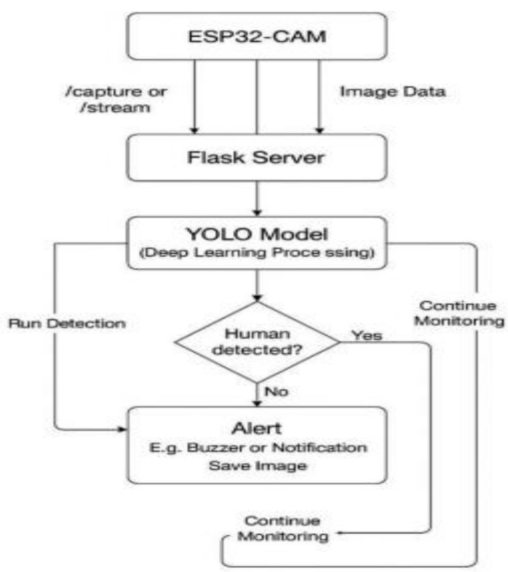

FIGURE 2. Flowchart of IOT based Smart Border Security System

The flowchart outlines the logical sequence in the IoT-Based Smart Border Security System, integrating the ESP32-CAM, a Flask server, and the YOLO human detection model. The ESP32-CAM continuously monitors its surroundings and either captures images or streams live video, sending this data to the connected Flask server. The Flask server functions as the backend, receiving the visual data and passing it to the YOLO model for human detection and further analysis. If a human is detected, appropriate alerts are triggered for security response. The YOLO (You Only Look Once) model is a deep learning-based real-time object detection system used in this project to identify human presence in captured video frames. When a person is detected, the system triggers an alert by activating a local buzzer, sending a remote notification, and saving the image for evidence. If no intruder is found, it continues to monitor the environment, looping continuously to process new incoming data for any potential threats.

Operation: The Smart Border Security System operates by continuously monitoring sensitive zones using the ESP32-CAM and AI-based detection. Each captured frame is sent to a Flask server, where the YOLO model checks for human presence. If a person is detected, the system triggers a buzzer for on-site alert, saves the image, and sends notifications via IoT platforms like Blynk or Telegram. If no intruder is found, it resumes monitoring. This real-time detection loop ensures constant surveillance, immediate alerts, and efficient response, making it ideal for high-security applications with features like low power usage, scalability, and accurate AI-based intrusion detection.

5. HARDWARE DESCRIPTION

ESP32-CAM Module: The ESP32-CAM is the core of this project, featuring a dual-core ESP32-S microcontroller with built-in Wi-Fi and Bluetooth. It includes an OV2640 camera sensor that captures images up to 2MP, supporting resolutions like 320×240, 640×480, and 1600×1200, enabling efficient image and video streaming for surveillance applications.

FIGURE 3. ESP32 Camera Module

The ESP32-CAM serves as both the camera and communication hub in the smart surveillance system. It captures images or streams video using its 2MP OV2640 camera and transmits data via Wi-Fi to a Flask server for analysis. With GPIO pins for connecting peripherals like buzzers or PIR sensors, and a microSD slot for local storage, it supports real-time monitoring and alerting. Powered by a dual-core ESP32-S microcontroller, the module combines image capture, IoT connectivity, and edge AI capabilities in a compact, cost-effective form ideal for security applications.

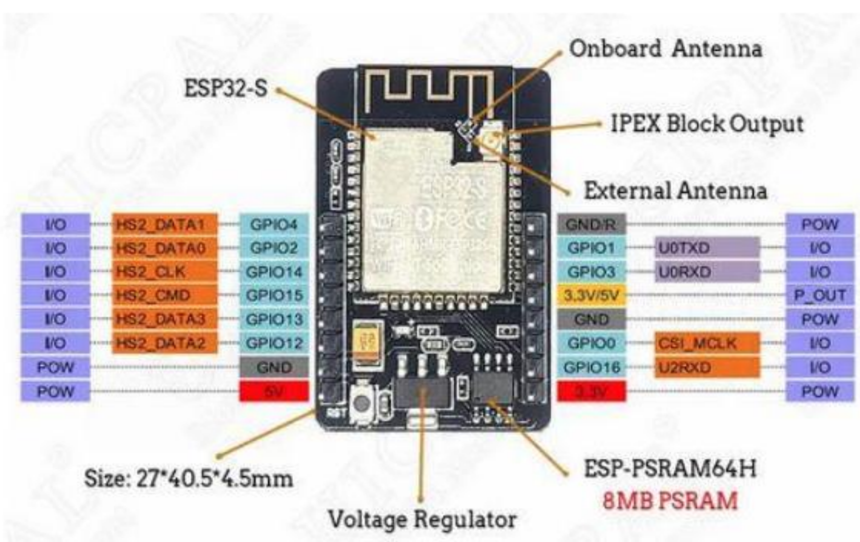

FIGURE 4. GPIO Pins of ESP 32 Camera Module

Recommended GPIOs for Custom Use:

FTDI Programmer Module (USB to Serial Converter): The ESP32-CAM does not have a built-in USB interface, so an external USB-to-Serial converter is needed for programming. The FTDI module serves this purpose by enabling serial communication between the ESP32-CAM and a computer during firmware upload. This module connects to the TX, RX, GND, and VCC pins of the ESP32-CAM. Additionally, to enable programming mode, the GPIO0 pin of the ESP32-CAM must be connected to GND during code upload.

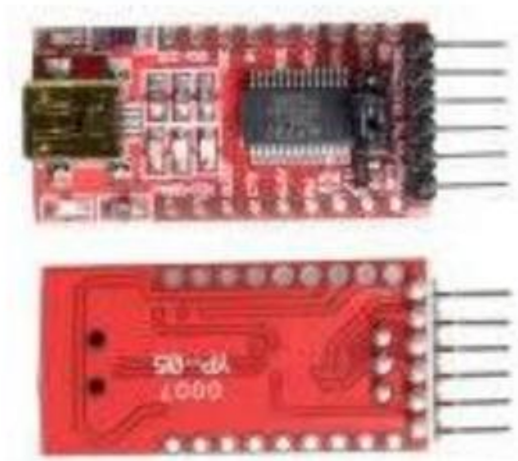

FIGURE 5. FTDI Programmer Module

Key Features of the FTDI Module: It uses the FT232RL chip to enable USB-to-serial (UART) communication, essential for programming devices like the ESP32-CAM. The module supports both 3.3V and 5V operation (selectable via jumper or solder pad), offers plug-and-play functionality, and is primarily used for flashing code onto the ESP32-CAM during development.

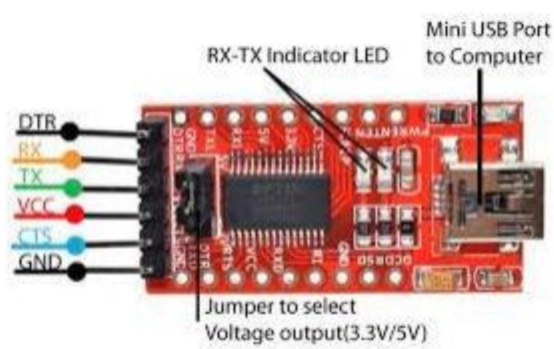

FIGURE 6. Pin Configuration of FTDI

Buzzer Module: The buzzer serves as an alert mechanism in the system. It is triggered whenever the YOLO object detection model identifies a human in the camera feed. This real-time audible alert provides immediate feedback about a detected intrusion, making it suitable for border or perimeter security.

FIGURE 7. Buzzer

Power Supply Description: The ESP32-CAM module needs a stable 5V DC power supply to function reliably, especially during Wi-Fi transmission, where current spikes can reach 300–500mA.
Power Supply Options:

- FTDI Module: Supplies 5V from a USB port during programming or testing; ensure it's set to 5V mode.
- External 5V Supply: For independent use, a regulated 5V adapter or a battery pack (like a 18650 Li-ion with a boost converter) is recommended.

FIGURE 8. Power Supply Source

Some ESP32-CAM boards include an onboard AMS1117 voltage regulator, which steps down 5V to the 3.3V required by the ESP32 chip.

Key Point:

- Always connect a 5V supply to the 5V pin of the ESP32-CAM to ensure stable operation, especially if the board relies on the onboard AMS1117 for voltage regulation.

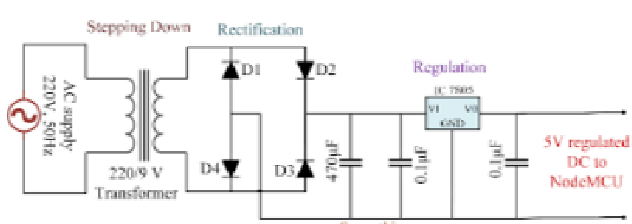

FIGURE 9. Circuit Diagram for Power Supply

Electronic components like the ESP32-CAM require a stable low-voltage DC supply, typically 5V. High-voltage AC from a wall socket (220V or 110V) is first stepped down using a transformer (e.g., to 12V AC), then converted to DC using a bridge rectifier. A capacitor smooths the resulting pulsating DC, and a voltage regulator (like 7805) ensures a steady 5V output. Additionally, never power the ESP32-CAM via the 3.3V pin as it can't provide enough current. Use a 100µF capacitor between 5V and GND to handle voltage spikes and ensure stable operation.

PIR Sensor (Passive Infrared Sensor):
The PIR sensor is used to detect motion or human presence by sensing infrared heat radiation. It acts as a trigger for the ESP32-CAM to start streaming or detecting.

FIGURE 10. PIR Sensor (Passive Infrared Sensor)

FIGURE 11. PIR Sensor Working Model

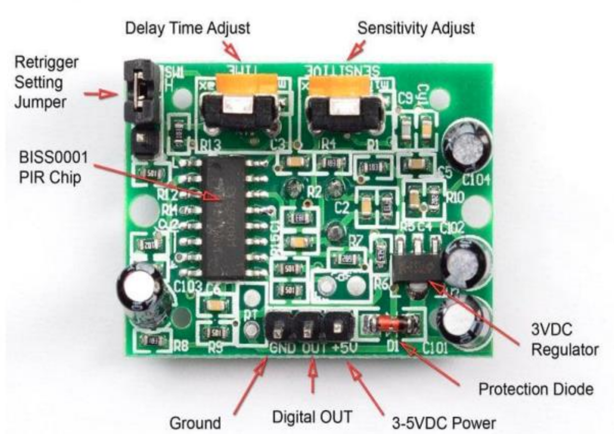

FIGURE 12. PIR Sensor Onboard Component Usage

The PIR sensor on-board components, such as sensitivity and delay adjustments, allow for customized motion detection. The sensitivity controls the detection range, with higher sensitivity detecting motion up to 6–7 meters, while lower sensitivity is for short-range detection. The delay time determines how long the output stays HIGH after motion is detected, with longer delays keeping alerts active for up to 5 minutes and shorter delays for brief alerts. The sensor can also be set to re-triggerable mode, where the delay resets with continuous motion, or non-re-triggerable mode, where further motion is ignored until the delay expires. For the Border Security System, high sensitivity and short delay settings can effectively detect intruders while minimizing false alerts.

6. SOFTWARE DESCRIPTION

Arduino IDE Setup: To set up the Arduino IDE, first download it from the official Arduino website (arduino.cc) for your operating system (Windows, macOS, or Linux) and install it. After installation, connect your Arduino board (e.g., Uno, Nano, or Mega) to your computer using a USB cable. Open the Arduino IDE, go to the "Tools" menu, select the appropriate board under "Board" and the correct port under "Port" (usually labeled as a COM port on Windows or a USB port on macOS). You can write your own code ("sketch") or open an example sketch under "File > Examples." Once your code is ready, click the "Upload" button to compile and transfer it to your Arduino board. If your sketch uses serial communication, open the "Serial Monitor" from the "Tools" menu to view output from the board. This setup allows you to start programming and interacting with your Arduino hardware.

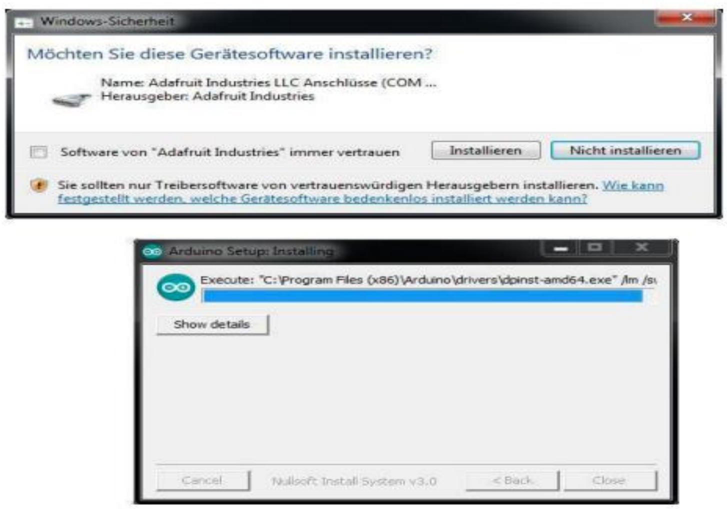

FIGURE 13. Installation of Arduino IDE - Driver

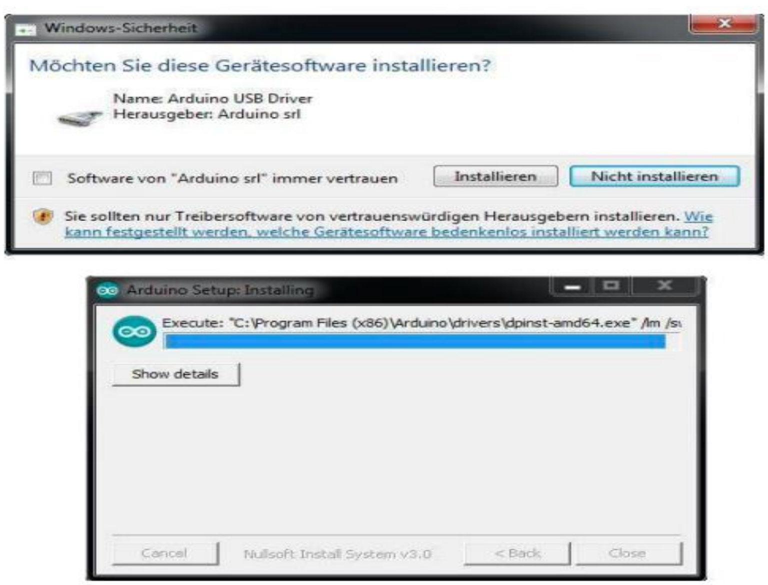

FIGURE 14. Installation of Arduino IDE - Other Drivers

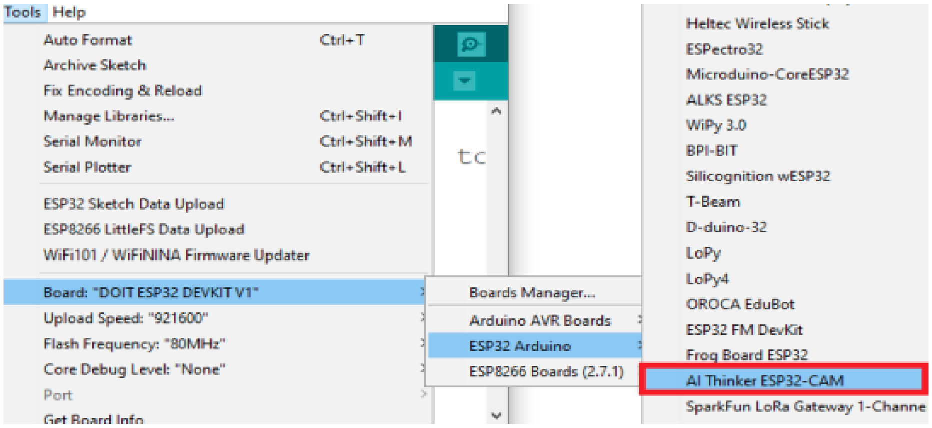

FIGURE 15. Board Manager SETUP

7. RESULTS

The IoT Based Smart Border Security System, developed using the ESP32-CAM module and the YOLO deep learning model with OpenCV in a Python Flask environment, successfully provided real-time video surveillance and alert capabilities. The system streamed video from the ESP32-CAM, processed frames with YOLO for object detection, and triggered an audible alarm upon detecting human presence. The Flask web interface displayed annotated video frames, and the system responded quickly to motion with minimal delay. Testing confirmed robust performance under various conditions, including different lighting scenarios and the presence of multiple people or objects, without false alarms. The system proved effective in border surveillance, offering high accuracy, fast response time, low power consumption, and a user-friendly interface for remote monitoring.

8. ADVANTAGES, DISADVANTAGES AND APPLICATIONS

The IoT Based Smart Border Security System offers real-time surveillance through the ESP32-CAM, providing cost-effective, compact, and wireless monitoring for remote areas. It utilizes the YOLO deep learning model for fast and accurate human detection, triggering alerts via a buzzer or IoT notifications. However, its limited camera range and reliance on stable Wi-Fi connectivity may pose challenges in large areas or remote locations with poor internet. The system has various applications, including border security, military zone monitoring, private property surveillance, smart city projects, wildlife protection, construction site monitoring, and remote installations like solar farms and telecom towers, offering scalable and efficient security solutions for diverse environments.

9. CONCLUSION

The IoT Based Smart Border Security System demonstrates the effective use of IoT and AI to enhance surveillance and intrusion detection at high-risk areas like national borders. By utilizing the ESP32-CAM module for live streaming and YOLO object detection for accurate human identification, the system triggers alerts via a buzzer and logs or sends event data remotely. The project is cost-effective, autonomous, and scalable, suitable for national security applications. Additionally, using Flask and Python in VS Code enables web-based monitoring, providing flexibility and accessibility, while paving the way for intelligent, automated security systems with minimal human involvement.

Future Scope:

Future enhancements for the system could include integration with cloud platforms like AWS or Google Cloud for footage storage and management, adding IR sensors for night vision, and implementing facial recognition or threat classification to distinguish authorized from unauthorized individuals. A mobile app could allow remote control and real-time alerts, while solar panels could enable off-grid operation. Integration with drones or PTZ cameras could expand surveillance coverage, and advanced alert systems via SMS, email, or IoT platforms like Blynk could improve response. Additionally, edge AI deployment on devices like the ESP32 could eliminate the need for external processing.

REFERENCES

[1]. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://arxiv.org/abs/1506.02640

[2]. Kumar, A., & Prakash, R. (2020). "Application of IoT in National Security and Border Surveillance", International Journal of Computer Science Trends and Technology (IJCST), Vol. 8.

[3]. G. Venkateshwarlu, G. Vallathan, Senthilkumar Meyyappan and Mohammed Abdul Nasar, "Microstrip Patch Slot Antenna with Enhanced Bandwidth and Multiple Parasitic Patches", IEEE Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), 2024.

[4]. Senthilkumar Meyyappan, Kalyan Kasturi, G. Vijaya Lakshmi, J. Srinija Reddy and K. Grace Sampoorna, "Improvement of LEACH Protocol for Enhancing Features of WSN", *Journal on Electronic and Automation Engineering*, Vol. 2(4), December 2023, pp. 19-26.

[5]. M. Senthil Kumar and Ashish Chaturvedi, "Energy-Efficient Coverage and Prolongs for Network Lifetime of WSN using MCP*", European Journal of Scientific Research (EJSR).* (Vol.95, No.2, January 2013, ISSN: 1450 – 216X / 1450 – 202X).

[6]. M. Senthil Kumar and Ashish Chaturvedi, "A Novel Enhanced Coverage Optimization Algorithm for Effectively Solving Energy Optimization Problem in WSN", *Research Journal of Applied Sciences, Engineering and Technology (RJASET).* (Issue 4, Vol.7, January 2014, ISSN: 2040 – 7459 & e-ISSN: 2040 – 7467).

[7]. Senthilkumar Meyyappan, G. Lava Kumar, G. Niharika and G. Chakradhar, "Cellular Network Signal Strength Analyser", *Journal on Electronic and Automation Engineering*, Vol. 4(1), March 2025, pp. 165-174.

[8]. M. Senthil Kumar and C. Sridhathan, "Impact of Mobility on the Routine of Enhanced – DSDV Protocol in Mobile Ad-hoc Networks", *International Journal of Applied Engineering Research (IJAER)*. (Vol.13, No.14, 2018, PP 11674-11679, ISSN: 0973-4562).

[9]. M. Senthil Kumar, "Energy Efficient Techniques for Transmission of Data in Wireless Sensor Networks", *Journal of Computing Technologies (JCT)*. (Vol.5, Issue 2, February 2016, ISSN: 2278 – 3814).

[10]. C.I. Vimalarani and M. Senthil Kumar, "Energy Efficient PCP Protocol for k-Coverage in Sensor Networks", *IEEE International Conference on Computational Intelligence and Computing Research, IEEE Proceedings, 2010.*

[11]. Senthilkumar Meyyappan, K. Susmitha, K. Vaishnavi and M. Sai Rao, "Condition Based Monitoring and Maintenance System for Underground Metro Stations", *Journal on Electronic and Automation Engineering*, Vol. 4(1), March 2025, pp. 175-182.

[12]. M. Senthil Kumar and M. Gopinath, "An Efficient Polynomial Pool-Based Scheme for Distributed Heterogeneous WSNs", *International Journal of Modern Engineering Research (IJMER).* (Vol.3, Issue 6, Nov-Dec.2013, PP 3328-3335, ISSN: 2249-6645).

[13]. M. Senthil Kumar and L. Praveen, "An Assuring Approach for Tree-Based Routing Topology in WSNs", *International Journal of Emerging Trends in Engineering and Development (IJETED).* (Issue 3, Vol.6, November 2013, ISSN: 2249 – 6149).

[14]. Senthilkumar Meyyappan and N. Selvamuthukumaran, "Network Selection in Heterogeneous Wireless Systems using GRA Method", *Journal on Electronic and Automation Engineering*, Vol. 4(1), March 2025, pp. 127-132.

[15]. M. Kavitha, T. Maheshwaran and M. Senthil Kumar, "Secure Routing in MANETs with Key Management", *International Journal on Engineering Technology and Sciences (IJETS).* (Vol.1, Issue 6, October 2014, ISSN (P): 2349 – 3968, ISSN (O): 2349 - 3976).

[16]. K. Arutselvan, C. Sridhathan and M. Senthil Kumar *"Unlocking Mobile Devices using Improved Face Recognition and Eye Blinking Technique",* International Journal of Applied Engineering Research (IJAER). (Vol.13, No.24, 2018, PP 16907-16909, ISSN: 0973-4562).

[17]. M. Kavitha, T. Maheshwaran and M. Senthil Kumar, "Ensure Data Transmission in Mobile Ad-Hoc Networks", *International Journal on Engineering Technology and Sciences (IJETS)*. (Vol.2, Issue 4, April 2015, ISSN (P): 2349 – 3968, ISSN (O): 2349 - 3976).

[18]. R. Kathiresh, V.M. Ramprasath and M. Senthil Kumar, *"A Systematic Approach for Design of Compressed Test Data in SOC"*, CiiT International Journal of Software Engineering and Technology. (Vol.4, No.4, Issue: April 2012, Print: ISSN 0974 – 9748 & Online: ISSN 0974 – 9632).

[19]. M. Senthil Kumar, T. Lokesh, T. Srikanth and T. Sowmya Goud, *"Enhancing Packet Inspection Accuracy to Identify Network Layer Attacks using Machine Learning",* International Journal of Scientific Research in Engineering and Management (IJSREM), Vol. 7, Issue 6, June 2023.

[20]. M. Subashinidevi, S. Castro and M. Senthil Kumar, "Efficient Anonymous Transfer of Data in Wireless Networks", *International Journal on Engineering Technology and Sciences (IJETS)*. (Vol.1, Issue 6, October 2014, ISSN (P): 2349 – 3968, ISSN (O): 2349 - 3976).