Gesture-Based Control System

Injamuri Sanjana Sri[1], Ashish Reddy Manne[2],Vanaparthi Sai Krishna[3]

[1,2,3]*Student, Department of Computer Science and Engineering, Anurag University, Hyderabad, India.*

21eg105j44@anurag.edu.in
21eg105j04@anurag.edu.in
21eg105j35@anurag.edu.in

Abstract. This paper presents a sensor-based approach to developing a gesture-controlled robotic vehicle, designed to enhance human-machine interaction in a more intuitive and accessible manner. The system translates hand movements, detected through accelerometers and gyroscopes within a wearable device, into commands that control the vehicle. Traditional methods for operating robotic systems, such as joysticks or remote controls, often require physical interaction, limiting their effectiveness, especially for individuals with disabilities or in environments where hands-free control is essential. This approach offers a non-contact, real-time control mechanism, detecting hand gestures without relying on visual cues, thereby reducing computational load and ensuring stable performance even under varying lighting conditions. The implementation involves sensor calibration, data acquisition, and processing via microcontrollers such as Arduino or ESP32, coupled with a motor control system for vehicle operation. Testing results demonstrate the accuracy and low-latency response of the gesture-based control system. The primary objective is to make this technology widely accessible, offering a scalable solution for various applications, including assistive technologies and robotics. Preliminary findings confirm the system's effectiveness, opening the door for further real-world applications.

Keywords. Gesture-based control, Robotic vehicle, Human-machine interaction, Wearable sensors, Assistive technology.

1 INTRODUCTION

Gesture control is one of the most promising approaches made from the broad field of human-machine interaction and offers an intuitive means for natural movements for people to interact with machines. The demand for gesture-controlled systems results from the constraints imposed by the traditional interfaces such as joysticks, remote controls, and voice commands, which are typically awkward in situations requiring quick and accurate movement or when hands-free interaction is needed. Besides, users with physical disabilities often face significant barriers to accessibility because of conventional systems of control.

In the past few years, advances in sensor technology have made it at all possible to overcome these problems because now it becomes technologically possible to build systems that could perceive hand movements and convert them into commands in real time. The most promising of these are accelerometers and gyroscopes because they can measure motion and orientation without needing to refer to external cameras or visual input. They are particularly ideal for applications where vision-based systems would be less reliable: for example, with low light, high visual noise, etc. The purpose of this study was to design a sensor-based gesture control system for robotic vehicles.

Unlike vision-based systems, where a complicated set of algorithms processes images, the motion sensor is used for the detection of hand movements. So, this system is less computationally intensive and robust. The sensors are integrated into a glove, so it detects the orientation and movement of the hand to control the movement of a vehicle. The simplicity, responsiveness in real-time, and accessibility have been emphasized in designing the system.

This work is the continuation of the rest of the studies conducted in the field of gesture control but with a different design, and the elimination of flex sensors and vision-based setups. It focuses solely on accelerometers and gyroscopes for its recognition process. This dimension reduces the problems that are inherent with visual recognition, such as light dependency and occlusion but maintains the same level of accuracy and responsiveness. Fundamental applications of such a system go beyond just vehicle control, like assistive devices for people with disabilities, smart home technologies, and industrial automation. However, this project is mainly focused on developing a robust, real-time system to be used in the control of a robotic vehicle. This vehicle has been implemented to move forward, reverse, or turn left or right by utilizing specific hand gestures by detecting through the sensors.

2 LITERATURE SURVEY

Studies have widely explored the development of gesture-based interfaces for controlling robotic vehicles or machines. A very prominent study by Zhang et al. (2017) entitled "Real- Time Hand Gesture Recognition Using Deep Learning" highlights the use of vision-centric systems to recognize gestures. However, the requirement of large datasets and substantial processing power is a limitation, particularly when such application is required to operate in real-time. Besides that, visual input-dependent systems are highly susceptible to diverse environmental conditions, such as illumination and occlusions, reducing their dependability in dynamic environments.

Another relevant paper, Rahul Swarup (2024), discussed vision-based smart car which recognizes hand gestures via image recognition to be controlled. Again, this also has the prospect of gesture control but, like other implementations, is plagued by issues concerning lighting conditions and processing power requirements that limit its scalability and practical use.

Acceleration and gyro-based methods are more reliable compared to those that require visual inputs.

These sensors can detect changes in orientation and acceleration, which are then matched to specific gestures. Since sensor-based approaches replace visual cues with physical data, they avoid many of the issues associated with vision-based recognition.

While a large portion of prior work has indeed explored hybrid approaches that leverage both vision and sensors, there is a surprising lack of focused research on sensor-based systems-an area on which this work primarily focuses.

This research aims to bridge this gap by creating the first sensor-driven gesture control system with simplified, yet robust and real-time execution of vision-based technologies at high precision levels. The central contribution of this paper therefore is in defining a system that can act robustly without requiring large datasets or computationally intensive algorithms.

3 RESEARCH METHODOLOGY

The development of the gesture-controlled robotic vehicle involves several stages that integrate both hardware and software components to achieve real-time control of the vehicle through hand gestures. The methodology is designed to provide a seamless interaction between the user’s hand movements and the robotic vehicle’s actions. This section outlines each step in the design and implementation process, detailing hardware setup, sensor calibration, algorithm development, wireless communication, and testing. The methodology ensures that every aspect of the system is reproducible and scalable for further applications.

3.1 System Architecture

The system architecture comprises two main components: the transmitter module (which the user wears to control the vehicle) and the receiver module (installed in the robotic vehicle). The primary goal is to detect hand gestures through sensor data and transmit commands wirelessly to the vehicle to execute the corresponding movements.

3.1.1 Transmitter Module (Wearable Device)

The wearable device, worn on the user's hand, consists of:

Arduino Nano: A small, versatile microcontroller used for processing sensor data in real time. The Nano collects data from the attached sensor and processes it to generate commands for controlling the vehicle.

MPU6050 Accelerometer and Gyroscope: This is a 6-DOF (degrees of freedom) sensor used for detecting hand movements. The accelerometer detects linear movements in x, y, and z axes, while the gyroscope detects rotational movements (angular velocity) around these axes.

nRF24L01+ Wireless Transceiver: A low-power wireless module that communicates with the receiver module. It operates at a 2.4 GHz frequency band and supports high-speed, low-latency data transmission.

Power Source: The transmitter module is powered by a 7.4V LiPo battery, ensuring portability and sufficient energy to power the components for extended periods.

The MPU6050 sensor continuously monitors the user’s hand movements, and the Arduino Nano processes this data to detect gestures. Once a gesture is detected, the corresponding control command is encoded and transmitted wirelessly using the nRF24L01+ module. The transmitter operates at intervals of 100 milliseconds to provide near real-time updates, ensuring that the vehicle responds instantly to the user's movements.

3.1.2 Receiver Module (Robotic Vehicle)

The receiver, mounted on the robotic vehicle, is responsible for interpreting the wireless commands sent from the transmitter and controlling the vehicle’s motors accordingly. It consists of:

Arduino Nano: Similar to the transmitter, the Arduino Nano is used here to receive the commands and process them to control the motors via the motor driver.

nRF24L01+ Wireless Transceiver: Paired with the transmitter, this module receives the control commands from the wearable device.

L298N Dual H-Bridge Motor Driver: This motor driver controls the movement of the vehicle’s four motors, allowing it to move forward, backward, turn left, or turn right, based on the received commands.

4WD Car Kit: The car chassis consists of four wheels, each powered by its own motor, providing stability and smooth movement across various terrains.

Power Source: The receiver and motors are powered by an 11.1V LiPo battery, ensuring sufficient power for the motors, driver, and wireless communication module.

The receiver's Arduino Nano processes the commands received from the transmitter via the nRF24L01+ module. Based on the commands, it activates the appropriate motors through the L298N motor driver to move the vehicle in the desired direction. This configuration allows for four basic actions: forward, backward, left turn, and right turn.

3.2 Sensor Design and Calibration

The accelerometers and gyroscopes, which are the main sensors utilized in this project, are embedded in a wearable glove. Accelerometers measure linear acceleration in three axes: X, Y, Z. Gyroscopes measure angular velocity. These sensors are calibrated to assure accurate readings of hand movements and orientations.

3.3 Vehicle Control

Once the gesture data is transmitted, the receiving Arduino Nano on the vehicle processes these commands to control movement. The vehicle's movements—such as moving forward, backward, or turning left and right—are governed by the logic coded into the receiver. This setup allows the vehicle to execute precise actions based on the gestures detected by the glove, effectively translating the user's hand movements into responsive vehicle control.

4. IMPLEMENTATION

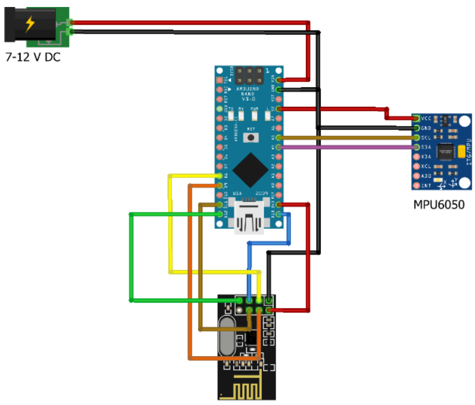
Here is the procedure of implementation: Sensor Selection and Calibration: Accelerometers and gyroscopes are selected for high accuracy and real-time responsiveness. Sensors are also calibrated to ensure proper and accurate readings about hand orientation and motion.

Microcontroller Configuration: Arduino is programmed to analyze the sensor data. The microcontroller interprets the unprocessed data from the sensors and correlates it with distinct gestures that are associated with vehicle operations.

Motor Driver Integration: Microcontroller and a motor driver, L298N drive this module. This module receives commands from the microcontroller and sends them to the vehicle motors for controlling the vehicle movement.

Wireless Communication: A nRF module has been added to the system to enable a communication between the glove and the vehicle. The integration will permit a user to operate a vehicle through a wireless medium. Testing and Debugging: The system undergo evaluation to confirm that the gesture recognition and vehicle control operate without interruption. Any problems, including latency in gesture identification or motor reactions, are resolved through debugging and optimization procedures.

4.1 Circuit Diagrams



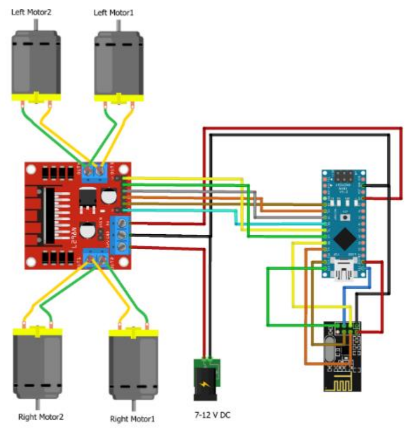


Figure 1 Transmitter System Circuit Diagram

The transmitter circuit diagram in Figure 1 consists of an Arduino Nano, an MPU6050 sensor, an nRF24L01+ wireless transceiver module, and a 7-12 V DC power source. The Arduino Nano acts as the main microcontroller, receiving data from the MPU6050, which is an accelerometer and gyroscope sensor module. The MPU6050 is connected to the Arduino Nano via I2C communication, allowing it to send orientation and movement data. The nRF24L01+ module is connected to the Arduino Nano for wireless transmission of data to a receiver module. Power is supplied by a 7-12 V DC battery, which powers both the Arduino and other components.

The receiver circuit diagram in Figure 2 consists of an Arduino Nano, an L298N motor driver, an nRF24L01+ wireless transceiver module, four DC motors, and a 7-12 V DC power source. The nRF24L01+ module receives data wirelessly from the transmitter circuit. The Arduino Nano processes the received signals and controls the movement of four motors connected through the L298N motor driver. Each motor is designated to control a wheel in a four-wheel drive (4WD) vehicle, allowing for precise movement based on the transmitted data. The power is provided by a 7-12 V DC battery, which supplies both the Arduino Nano and the motor driver.

4.2 Algorithm

4.2.1 Transmitter System

Steps:

1. **Initialization**
   - 1.1. Initialize I2C communication for the MPU6050 sensor
   - 1.2. Initialize the nRF24L01+ RF module in transmitter mode, setting the transmission rate to 250 kbps and defining the output pipe.
   - 1.3. Set initial x-axis and y-axis values to 127 (neutral position).
2. **MPU6050 Configuration**
   - 2.1. Calibrate the MPU6050 by configuring the accelerometer and gyroscope offsets.
   - 2.2. Enable the Digital Motion Processing (DMP) on the MPU6050 to obtain motion data.
   - 2.3. Set the expected DMP packet size for future operations.
3. **Main Loop Execution**
   - 3.1. Check if the DMP is ready, ensuring that the MPU6050 is functional.
   - 3.2. Read the latest packet from the MPU6050's FIFO buffer.
   - 3.3. Extract quaternion data from the buffer and compute yaw, pitch, and roll.
   - 3.4. Constrain yaw and pitch values within a -90 to +90 degree range.
   - 3.5. Map the constrained yaw and pitch values into a range from 0 to 254 for transmission.
   - 3.6. Transmit the mapped x-axis and y-axis values wirelessly using the nRF24L01+ RF module.

4.2.2 Receiver System

Steps:

1. **Initialization**
   - 1.1. Initialize the nRF24L01+ RF module in receiver mode, setting the data rate to 250 kbps and defining the input pipe.
   - 1.2. Configure the GPIO pins for the left and right motor controls (direction and speed).
   - 1.3. Set initial motor speed to zero.
2. **Main Loop Execution**
   - 2.1. Continuously check for incoming data from the RF module.
   - 2.2. If a signal is received:
     - 2.2.1. Read the x-axis and y-axis values.
     - 2.2.2. Map the x-axis and y-axis values from the range of 0 to 254 to motor control values in the range of -255 to +255.
     - 2.2.3. Set the motor direction based on the sign of the y-axis value.
     - 2.2.4. Calculate the speed of the left and right motors based on the combined x-axis and y-axis values.
     - 2.2.5. Constrain the motor speeds within the range of 0 to 255.
     - 2.2.6. Update the motors to rotate with the computed speed and direction.
     - 2.2.7. Record the time of the last valid signal.
   - 2.3. If no signal is received within a predefined timeout period (e.g., 500 ms), stop both motors.

4.3 Working

There are four hand gestures which can be recognized by the car. They are FORWARD, BACKWARD , RIGHT,LEFT.

The following are the hand gestures used in controlling the car.

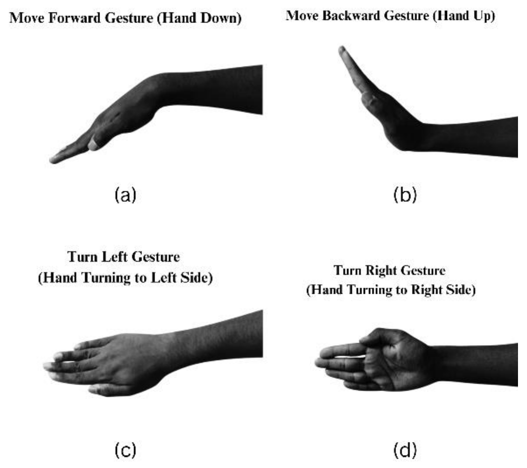


Figure 3 Hand Gesture Representations for Movement Control

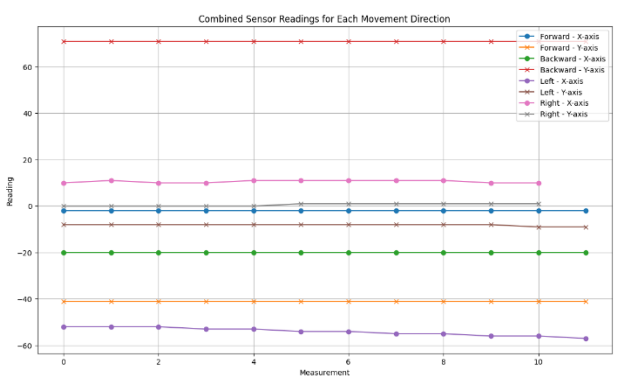
Figure 3 displays four distinct hand gestures used to control movement directions in a gesture-controlled system. Each gesture corresponds to a specific command:

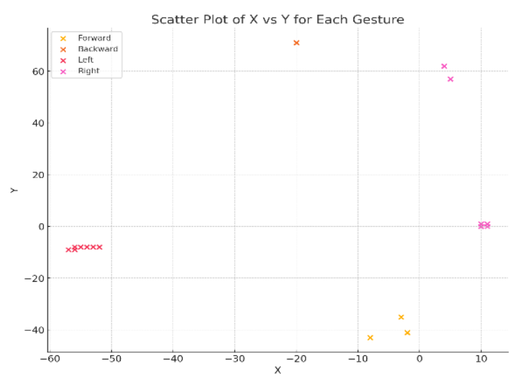
- **(a) Move Forward Gesture (Hand Down)**: The hand is positioned horizontally with the palm facing down, indicating the command to move forward.
- **(b) Move Backward Gesture (Hand Up)**: The hand is positioned vertically with the palm facing outward, signaling the command to move backward.
- **(c) Turn Left Gesture (Hand Turning to Left Side)**: The hand is rotated to the left, with the palm facing downward, representing the command to turn left.
- **(d) Turn Right Gesture (Hand Turning to Right Side)**: The hand is rotated to the right, with the palm facing downward, denoting the command to turn right.

These gestures serve as intuitive controls, allowing users to communicate movement directions in a gesture-based interface. The gestures are designed to be simple and distinct, reducing the likelihood of misinterpretation by the gesture-recognition system.

4.4 Final Readings

The system was tested in varying conditions to assess performance in terms of accuracy, responsiveness, and user-friendliness. The final readings were read as follows:





Responsiveness: The delay between gesture input and vehicle response was measured to be less than 200ms, so the system was very responsive.

The system was subjected to testing by several users and garnered considerable feed from them about ease of usability and precision in control.

5 CONCLUSION

The feasibility of a sensor-based gesture-controlled robotic vehicle system that does not rely on vision recognition or flex sensors has been successfully demonstrated in this research. The proposed method based on accelerometers and gyroscopes provides a reliable real-time method for controlling movement of a vehicle by hand gestures. The accuracy and responsiveness of the system are high, thus making it viable for usage in assistive technologies, robotics, and control of a smart vehicle. This will in turn lead to further functionality development within the system as well as more advanced refinement of the gesture recognition algorithm toward the ability to recognize complicated gestures in the future.

REFERENCES

1. Kumar, T. V. (2018). Project Risk Management System Development Based on Industry 4.0 Technology and its Practical Implications.
2. Tambi, V. K., & Singh, N. (2015). Potential Evaluation of REST Web Service Descriptions for Graph-Based Service Discovery with a Hypermedia Focus.
3. Kumar, T. V. (2024). A Comparison of SQL and NO-SQL Database Management Systems for Unstructured Data.
4. Kumar, T. V. (2024). A Comprehensive Empirical Study Determining Practitioners' Views on Docker Development Difficulties: Stack Overflow Analysis.
5. Kumar, T. V. (2024). Developments and Uses of Generative Artificial Intelligence and Present Experimental Data on the Impact on Productivity Applying Artificial Intelligence that is Generative.
6. Kumar, T. V. (2024). A New Framework and Performance Assessment Method for Distributed Deep Neural NetworkBased Middleware for Cyberattack Detection in the Smart IoT Ecosystem.
7. Sharma, S., & Dutta, N. (2016). Analysing Anomaly Process Detection using Classification Methods and Negative Selection Algorithms.
8. Sharma, S., & Dutta, N. (2024). Examining ChatGPT's and Other Models' Potential to Improve the Security Environment using Generative AI for Cybersecurity.
9. Sakshi, S. (2023). Development of a Project Risk Management System based on Industry 4.0 Technology and its Practical Implications.
10. Arora, P., & Bhardwaj, S. Mitigating the Security Issues and Challenges in the Internet of Things (IOT) Framework for Enhanced Security.
11. Sakshi, S. (2024). A Large-Scale Empirical Study Identifying Practitioners' Perspectives on Challenges in Docker Development: Analysis using Stack Overflow.
12. Sakshi, S. (2023). Advancements and Applications of Generative Artificial Intelligence and show the Experimental Evidence on the Productivity Effects using Generative Artificial Intelligence.
13. Sakshi, S. (2023). Assessment of Web Services based on SOAP and REST Principles using Different Metrics for Mobile Environment and Multimedia Conference.
14. Sakshi, S. (2022). Design and Implementation of a Pattern-based J2EE Application Development Environment.
15. Sharma, S., & Dutta, N. (2018). Development of New Smart City Applications using Blockchain Technology and Cybersecurity Utilisation. Development, 7(11).
16. Sharma, S., & Dutta, N. (2017). Development of Attractive Protection through Cyberattack Moderation and Traffic Impact Analysis for Connected Automated Vehicles. Development, 4(2).
17. Sharma, S., & Dutta, N. (2015). Evaluation of REST Web Service Descriptions for Graph-based Service Discovery with a Hypermedia Focus. Evaluation, 2(5).
18. Sharma, S., & Dutta, N. (2024). Examining ChatGPT's and Other Models' Potential to Improve the Security Environment using Generative AI for Cybersecurity.

19. Sharma, S., & Dutta, N. (2015). Cybersecurity Vulnerability Management using Novel Artificial Intelligence and Machine Learning Techniques. Sakshi, S. (2023). Development of a Project Risk Management System based on Industry 4.0 Technology and its Practical Implications.
20. Sharma, S., & Dutta, N. (2017). Classification and Feature Extraction in Artificial Intelligence-based Threat Detection using Analysing Methods.
21. Sharma, S., & Dutta, N. (2016). Analysing Anomaly Process Detection using Classification Methods and Negative Selection Algorithms.
22. Sharma, S., & Dutta, N. (2015). Distributed DNN-based Middleware for Cyberattack Detection in the Smart IOT Ecosystem: A Novel Framework and Performance Evaluation Technique.
23. Bhat, S. (2015). Technology for Chemical Industry Mixing and Processing. Technology, 2(2).
24. Bhat, S. (2024). Building Thermal Comforts with Various HVAC Systems and Optimum Conditions.
25. Bhat, S. (2020). Enhancing Data Centre Energy Efficiency with Modelling and Optimisation of End-To-End Cooling.
26. Bhat, S. (2016). Improving Data Centre Energy Efficiency with End-To-End Cooling Modelling and Optimisation.
27. Bhat, S. (2015). Deep Reinforcement Learning for Energy-Saving Thermal Comfort Management in Intelligent Structures.
28. Bhat, S. (2015). Design and Function of a Gas Turbine Range Extender for Hybrid Vehicles.
29. Bhat, S. (2023). Discovering the Attractiveness of Hydrogen-Fuelled Gas Turbines in Future Energy Systems.
30. Bhat, S. (2019). Data Centre Cooling Technology's Effect on Turbo-Mode Efficiency.
31. Bhat, S. (2018). The Impact of Data Centre Cooling Technology on Turbo-Mode Efficiency.
32. Archana, B., & Sreedaran, S. (2023). Synthesis, characterization, DNA binding and cleavage studies, in-vitro antimicrobial, cytotoxicity assay of new manganese (III) complexes of N-functionalized macrocyclic cyclam based Schiff base ligands. Polyhedron, 231, 116269.
33. Archana, B., & Sreedaran, S. (2022). New cyclam based Zn (II) complexes: effect of flexibility and para substitution on DNA binding, in vitro cytotoxic studies and antimicrobial activities. Journal of Chemical Sciences, 134(4), 102.
34. Archana, B., & Sreedaran, S. (2021). POTENTIALLY ACTIVE TRANSITION METAL COMPLEXES SYNTHESIZED AS SELECTIVE DNA BINDING AND ANTIMICROBIAL AGENTS. European Journal of Molecular and Clinical Medicine, 8(1), 1962-1971.
35. Rasappan, A. S., Palanisamy, R., Thangamuthu, V., Dharmalingam, V. P., Natarajan, M., Archana, B., ... & Kim, J. (2024). Battery-type WS2 decorated WO3 nanorods for high-performance supercapacitors. Materials Letters, 357, 135640.
36. Arora, P., & Bhardwaj, S. (2017). Investigation and Evaluation of Strategic Approaches Critically before Approving Cloud Computing Service Frameworks.
37. Arora, P., & Bhardwaj, S. (2017). Enhancing Security using Knowledge Discovery and Data Mining Methods in Cloud Computing.
38. Arora, P., & Bhardwaj, S. (2017). Combining Internet of Things and Wireless Sensor Networks: A Security-based and Hierarchical Approach.
39. Arora, P., & Bhardwaj, S. (2019). Safe and Dependable Intrusion Detection Method Designs Created with Artificial Intelligence Techniques. machine learning, 8(7).
40. Arora, P., & Bhardwaj, S. (2017). A Very Safe and Effective Way to Protect Privacy in Cloud Data Storage Configurations.
41. Arora, P., & Bhardwaj, S. (2019). The Suitability of Different Cybersecurity Services to Stop Smart Home Attacks.
42. Arora, P., & Bhardwaj, S. (2020). Research on Cybersecurity Issues and Solutions for Intelligent Transportation Systems.
43. Arora, P., & Bhardwaj, S. (2021). Methods for Threat and Risk Assessment and Mitigation to Improve Security in the Automotive Sector. Methods, 8(2).

44. Onyema, E. M., Gude, V., Bhatt, A., Aggarwal, A., Kumar, S., Benson-Emenike, M. E., & Nwobodo, L. O. (2023). Smart Job Scheduling Model for Cloud Computing Network Application. *SN Computer Science*, *5*(1), 39.
45. Hasnain, M., Gude, V., Edeh, M. O., Masood, F., Khan, W. U., Imad, M., & Fidelia, N. O. (2024). Cloud-Enhanced Machine Learning for Handwritten Character Recognition in Dementia Patients. In *Driving Transformative Technology Trends With Cloud Computing* (pp. 328-341). IGI Global.
46. Kumar, M. A., Onyema, E. M., Sundaravadivazhagan, B., Gupta, M., Shankar, A., Gude, V., & Yamsani, N. (2024). Detection and mitigation of few control plane attacks in software defined network environments using deep learning algorithm. *Concurrency and Computation: Practice and Experience*, *36*(26), e8256.
47. Gude, V., Lavanya, D., Hameeda, S., Rao, G. S., & Nidhya, M. S. (2023, December). Activation of Sleep and Active Node in Wireless Sensor Networks using Fuzzy Logic Routing Table. In *2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)* (pp. 1358-1360). IEEE.
48. Gorantla, V. A. K., Sriramulugari, S. K., Gorantla, B., Yuvaraj, N., & Singh, K. (2024, March). Optimizing performance of cloud computing management algorithm for high-traffic networks. In *2024 2nd International Conference on Disruptive Technologies (ICDT)* (pp. 482-487). IEEE.
49. Sriramulugari, S. K., & Gorantla, V. A. K. (2023). Deep learning based convolutional geometric group network for alzheimer disease prediction. *International Journal of Biotech Trends and Technology*, *13*(3).
50. Sriramulugari, S. K., & Gorantla, V. A. K. Cyber Security using Cryptographic Algorithms.
51. Gorantla, V. A. K., Sriramulugari, S. K., Mewada, A. H., Jiwani, N., & Kiruthiga, T. (2023, December). The slicing based spreading analysis for melanoma prediction using reinforcement learning model. In *2023 IEEE Technology & Engineering Management Conference-Asia Pacific (TEMSCON-ASPAC)* (pp. 1-7). IEEE.
52. Sriramulugari, S. K., Gorantla, V. A. K., Mewada, A. H., Gupta, K., & Kiruthiga, T. (2023, December). The opinion based analysis for stressed adults using sentimental mining model. In *2023 IEEE Technology & Engineering Management Conference-Asia Pacific (TEMSCON-ASPAC)* (pp. 1-6). IEEE.
53. Gorantla, V. A. K., Sriramulugari, S. K., Mewada, A. H., Gupta, K., & Kiruthiga, T. (2023, December). The smart computation of multi-organ spreading analysis of COVID-19 using fuzzy based logical controller. In *2023 IEEE Technology & Engineering Management Conference-Asia Pacific (TEMSCON-ASPAC)* (pp. 1-7). IEEE.
54. Gude, Venkataramaiah (2023). Machine Learning for Characterization and Analysis of Microstructure and Spectral Data of Materials. *International Journal of Intelligent Systems and Applications in Engineering* 12 (21):820 - 826.
55. Prabhu Kavin, B., Karki, S., Hemalatha, S., Singh, D., Vijayalakshmi, R., Thangamani, M., ... & Adigo, A. G. (2022). Machine Learning-Based Secure Data Acquisition for Fake Accounts Detection in Future Mobile Communication Networks. *Wireless Communications and Mobile Computing*, *2022*(1), 6356152.
56. Kalaiselvi, B., & Thangamani, M. (2020). An efficient Pearson correlation based improved random forest classification for protein structure prediction techniques. *Measurement*, *162*, 107885.
57. Thangamani, M., Satheesh, S., Lingisetty, R., Rajendran, S., & Shivahare, B. D. (2025). Mathematical Model for Swarm Optimization in Multimodal Biomedical Images. In *Swarm Optimization for Biomedical Applications* (pp. 86-107). CRC Press.
58. Chithrakumar, T., Mathivanan, S. K., Thangamani, M., Balusamy, B., Gite, S., & Deshpande, N. (2024, August). Revolutionizing Agriculture through Cyber Physical Systems: The Role of Robotics in Smart Farming. In *2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT)* (Vol. 1, pp. 1-6). IEEE.
59. Tiwari, V., Ananthakumaran, S., Shree, M. R., Thangamani, M., Pushpavalli, M., & Patil, S. B. (2024). RETRACTED ARTICLE: Data analysis algorithm for internet of things based on federated learning with optical technology. *Optical and Quantum Electronics*, *56*(4), 572.

60. Sakthivel, M., SivaSubramanian, S., Prasad, G. N. R., & Thangamani, M. (2023). Automated detection of cardiac arrest in human beings using auto encoders. Measurement: Sensors, 27, 100792.
61. CHITHRAKUMAR, T., THANGAMANI, M., KSHIRSAGAR, R. P., & JAGANNADHAM, D. (2023). MICROCLIMATE PREDICTION USING INTERNET OF THINGS (IOT) BASED ENSEMBLE MODEL. *Journal of Environmental Protection and Ecology*, *24*(2), 622-631.
62. Vasista, T. G. K. (2017). Towards innovative methods of construction cost management and control. *Civ Eng Urban Plan: Int J*, *4*, 15-24.
63. Hsu, H. Y., Hwang, M. H., & Chiu, Y. S. P. (2021). Development of a strategic framework for sustainable supply chain management. *AIMS Environmental Science*, (6).
64. Venkateswarlu, M., & Vasista, T. G. (2023). Extraction, Transformation and Loading Process in the Cloud computing scenario. *International Journal of Engineering Applied Sciences and Technology*, *8*, 232-236.
65. Sagar, M., & Vanmathi, C. (2022, August). Network Cluster Reliability with Enhanced Security and Privacy of IoT Data for Anomaly Detection Using a Deep Learning Model. In *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)* (pp. 1670-1677). IEEE.
66. Sagar, M., & Vanmathi, C. (2024). A Comprehensive Review on Deep Learning Techniques on Cyber Attacks on Cyber Physical Systems. *SN Computer Science*, *5*(7), 891.
67. Sagar, M., & Vanmathi, C. (2024). Hybrid intelligent technique for intrusion detection in cyber physical systems with improved feature set. *Journal of Intelligent & Fuzzy Systems*, (Preprint), 1-17.
68. Vanmathi, C., Mangayarkarasi, R., Prabhavathy, P., Hemalatha, S., & Sagar, M. (2023). A Study of Human Interaction Emotional Intelligence in Healthcare Applications. In *Multidisciplinary Applications of Deep Learning-Based Artificial Emotional Intelligence* (pp. 151-165). IGI Global.
69. Kumar, N. A., & Kumar, J. (2009). *A Study on Measurement and Classification of TwitterAccounts*.
70. Senthilkumar, S., Haidari, M., Devi, G., Britto, A. S. F., Gorthi, R., & Sivaramkrishnan, M. (2022, October). Wireless bidirectional power transfer for E-vehicle charging system. In *2022 International Conference on Edge Computing and Applications (ICECAA)* (pp. 705-710). IEEE.
71. Firos, A., Prakash, N., Gorthi, R., Soni, M., Kumar, S., & Balaraju, V. (2023, February). Fault detection in power transmission lines using AI model. In *2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS)* (pp. 1-6). IEEE.
72. Gorthi, R. S., Babu, K. G., & Prasad, D. S. S. (2014). Simulink model for cost-effective analysis of hybrid system. *International Journal of Modern Engineering Research (IJMER)*, *4*(2).
73. Rao, P. R., & Sucharita, D. V. (2019). A framework to automate cloud based service attacks detection and prevention. *International Journal of Advanced Computer Science and Applications*, *10*(2), 241-250.
74. Rao, P. R., Sridhar, S. V., & RamaKrishna, V. (2013). An Optimistic Approach for Query Construction and Execution in Cloud Computing Environment. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(5).
75. Rao, P. R., & Sucharita, V. (2020). A secure cloud service deployment framework for DevOps. *Indonesian Journal of Electrical Engineering and Computer Science*, *21*(2), 874-885.
76. Selvan, M. A., & Amali, S. M. J. (2024). RAINFALL DETECTION USING DEEP LEARNING TECHNIQUE.