iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: III Month of publication: March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78311

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

Real-Time Two-Wheeler Accident Detection with Automated Emergency Notification and Post-Impact Voice Logging Using IoT

Tushar Bhadane[1], Arnav Abhigya[2], Neel Chotalia[3], Nandit Aghera[4]
[1, 2, 3, 4]*Electronics & Communication Engineering, Vishwakarma Government Engineering College, Chandkheda (382424), Gujarat, India*

Abstract: Two-wheeler riders are disproportionately vulnerable to severe road accident injuries due to the absence of automated emergency response mechanisms. This paper presents a real-time accident detection and emergency notification system designed specifically for motorcycles, leveraging Internet of Things (IoT) technologies. The proposed system employs an MPU6050 inertial measurement unit interfaced with an ESP32 microcontroller to continuously monitor linear acceleration and angular motion at 100 Hz. A time-based threshold filtering algorithm distinguishes genuine collision events and rollovers from common false-trigger scenarios such as potholes, sharp turns, and hard braking — by requiring hazardous sensor readings to persist beyond calibrated temporal gates (≥ 100ms for impacts at > 7g; ≥ 500ms for tilt angles exceeding ±75°). Upon accident confirmation, the system autonomously retrieves GPS coordinates via a NEO-6M module and dispatches location-embedded emergency SMS alerts through a SIM800L GSM module, while simultaneously uploading real-time data to the Blynk IoT cloud platform. A distinguishing feature of the system is its post-impact voice logging capability, wherein a MAX9814 microphone records 30–60 seconds of ambient audio, stored in WAV format on a MicroSD card to support forensic investigation and victim assessment. Experimental evaluation across ten simulated accident scenarios demonstrated reliable collision detection with a low false-alarm rate and average SMS delivery times of 8–12 seconds. The system's internet-independent architecture, cost-effective hardware, and scalable design make it a practical solution for improving emergency response times and enhancing safety across the global two-wheeler population.

Keywords: Accident Detection, IoT, ESP32, MPU6050, GPS Tracking, GSM, Voice Logging, Two-Wheeler Safety, Emergency Response.

I. INTRODUCTION

Two-wheeler riders face a higher risk of severe injury in road accidents due to limited physical protection and the absence of automated emergency response mechanisms [1]. In many cases, accident victims are unable to communicate their situation, resulting in delayed rescue and medical assistance.

To address these challenges, this project proposes an adaptive collision detection and emergency alert system tailored for motorcycle applications. The system employs an MPU6050 sensor unit to capture acceleration and angular motion, while an ESP32 processing unit executes adaptive thresholding and time-based filtering to accurately distinguish genuine collision events from false triggers such as potholes or hard braking. Emergency communication is achieved through a SIM800L GSM module and NEO-6M GPS module, enabling automatic transmission of location-based alerts. Additionally, a microphone and SD card module allow post-accident voice recording, and cloud connectivity enables real-time data logging and analysis.

The proposed system focuses on reducing response time, improving detection reliability, and providing actionable information to emergency responders, offering a cost-effective and scalable solution for enhancing two-wheeler safety.

II. LITERATURE REVIEW

The available literature on automated vehicle accident detection reveals a significant shift from manual reporting toward integrated IoT solutions designed to minimize emergency response times. Traditional systems, which rely heavily on eyewitness accounts and post-incident analysis, are increasingly viewed as unreliable, particularly in remote regions or during nighttime. To address this, research has focused on the synergy between microcontrollers and motion sensors. Early implementations utilized basic Arduino frameworks paired with MPU6050 accelerometers and GPS modules to transmit coordinate-based SMS alerts via GSM [3], [4].

However, as the field has evolved, more sophisticated hardware like the ESP32 and Raspberry Pi has been adopted to handle complex tasks. For instance, studies have explored specialized safety mechanisms for motorcyclists, including smart helmets and eye-blink monitoring to detect drowsiness [5], while others have integrated computer vision via OpenCV to provide an additional layer of verification through camera-based impact detection.

A recurring theme in the literature is the critical need for system reliability and the reduction of false alarms. Because two-wheelers frequently encounter non-accident vibrations from potholes or sharp turns, recent advancements have introduced machine learning models on edge devices to distinguish between genuine collisions and everyday riding irregularities [6], [7], [8]. Furthermore, the communication architecture of these systems has branched into two distinct paths: cloud-integrated platforms and standalone units. While some researchers advocate for using IoT platforms like ThingSpeak for real-time data logging and remote monitoring via Android applications [2], others emphasize the necessity of internet-independent systems. These standalone solutions prioritize direct GSM alerting sending precise latitude, longitude, and altitude data to ensure functionality in diverse geographical terrains with poor network coverage. Despite these advancements, a gap remains for a truly robust solution that combines reliable sensor fusion with post-impact features like voice logging to assist in forensic investigation and victim assessment.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. *System Overview*

The proposed system is powered by a 3.7V 18650 LiPo battery via a TP4056 charger and step-up boost module providing regulated 5V. The ESP32 (TTGO T-Call variant with integrated SIM800L) serves as the central processing unit. Table I summarizes the key hardware components.

TABLE I. Key Hardware Components and Functions

Component	Specification	Function
ESP32 (TTGO T-Call)	Dual-core 240MHz, Wi-Fi, integrated SIM800L	Central processing, communication hub
MPU6050	3-axis accel. + 3-axis gyro, I2C interface	Collision and rollover detection
Neo-6M GPS	L1 band, 22-channel, 3m accuracy	Real-time geographic coordinates
SIM800L GSM	Quad-band GSM/GPRS, up to 2A peak	GSM-SMS emergency alert
MAX9814 Mic	Auto Gain Control, 40dB SNR	Post-impact voice logging
MicroSD Module	SPI interface, FAT32 format	Local audio and event log storage
TP4056 + Boost	3.7V LiPo charge + 5V regulated output	Battery management and regulation

B. *System Block Diagram*

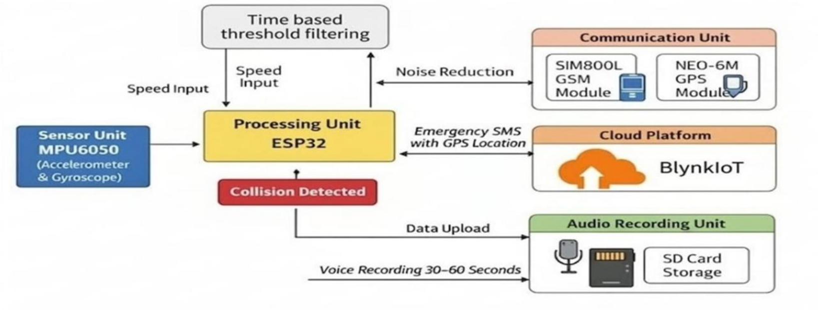


Fig. 1 Block diagram of the proposed system, illustrating data flow from the MPU6050 sensor unit through the ESP32 processing unit.

C. Algorithm and Operational Workflow

1) Time-Based Threshold Filtering Algorithm

Raw acceleration data from the MPU6050 is inherently susceptible to short-duration spikes caused by everyday riding irregularities — potholes, speed bumps, railway crossings, and sudden braking. A naive single-sample threshold check would misclassify these transient shocks as collision events. To address this, the system implements a Time-Based Threshold Filtering approach, requiring a hazardous condition to persist beyond a defined minimum duration before an accident is confirmed.

a) Sensor Data Acquisition: The MPU6050 is polled continuously at 100 Hz via I²C, delivering raw linear acceleration along three axes — Ax (lateral), Ay (longitudinal), and Az (vertical) in units of gravitational acceleration (g), alongside gyroscope angular velocity readings (Gx, Gy, Gz) in °/s.

b) Resultant G-Force Computation: At each sampling interval, the ESP32 computes the net acceleration magnitude using the Euclidean norm:

$$A\_net = \sqrt{(Ax^2 + Ay^2 + Az^2)} \ldots (1)$$

Under normal riding conditions, A_net approximates 1g due to gravity's contribution along the vertical axis. During a collision or rollover, simultaneous deviations across all three axes cause A_net to spike sharply, ensuring that oblique impacts — which may not register strongly on any single axis — are reliably captured.

c) Primary Threshold Check: A_net is compared against an empirically calibrated impact threshold T_impact of 7g. Exceeding this threshold does not immediately confirm an accident; instead, it initiates a temporal monitoring window.

d) Time-Based Duration Validation: A transient shock (e.g., a pothole) typically generates a spike that resolves within 30–50ms. A genuine collision sustains elevated sensor readings for a longer period. The algorithm therefore requires the threshold exceedance to persist continuously for at least 100ms. An internal counter increments with each qualifying sample and resets if the condition breaks before this gate is met. Only upon satisfying this duration condition is a Collision Detected flag raised.

e) Rollover and Tilt Detection: In parallel, gyroscope data is integrated over time to estimate the vehicle's roll angle θ. If θ exceeds ±75° from the nominal upright position and this tilt persists for ≥ 500ms-distinguishing a rollover from an aggressive cornering lean-a Rollover Detected flag is independently raised, triggering the same alert protocol.

f) Manual Override: A dedicated hardware panic button allows the rider to immediately bypass all sensor logic and raise the alert flag directly, accommodating scenarios where sensor readings may be inconclusive.

g) Re-Trigger Lockout: Once an alert is dispatched, a software lockout flag prevents duplicate SMS alerts within the same accident event. The lockout remains active until the system is manually reset via the hardware reset button or a remote Blynk command.

Decision Logic Summary: -

- D = True (Collision) → A_net > 7g sustained for ≥ 100ms
- D = True (Rollover) → θ > ±75° sustained for ≥ 500ms
- D = True (Manual) → Panic Button pressed
- D = False → All other conditions; monitoring continues

2) Operational Workflow Steps

a) System Initialization: On power-up, the ESP32 initializes all hardware modules sequentially: the MPU6050 is calibrated over I²C; the NEO-6M GPS module begins satellite acquisition over UART; the SIM800L GSM module registers on the cellular network; the MAX9814 microphone and MicroSD module are initialized over SPI; and the Blynk IoT connection is established over Wi-Fi. Upon successful initialization of all modules, the system enters Continuous Monitoring Mode.

b) Continuous Sensor Monitoring: The main loop polls the MPU6050 at 100 Hz, computing A_net and updating the roll angle θ on each cycle. Simultaneously, NEO-6M are parsed via TinyGPS++ to maintain a continuously updated GPS fix. Normal riding conditions (A_net: 1–3g, stable θ) keep the system in monitoring mode indefinitely.

c) Each sampling cycle checks: Whether A_net exceeds 7g or θ exceeds ±75°. If neither condition is met, the system continues monitoring. If a threshold is breached, the system tracks how long it persists. An accident is confirmed only if the breach lasts beyond the minimum required duration; a shorter breach resets the check.

d) GPS Coordinate Retrieval: Upon accident confirmation, the system retrieves the most recent valid GPS fix. If satellite lock is unavailable due to environmental obstruction, the last successfully recorded coordinates serve as a fallback, ensuring location data is always included in the alert.

e) Emergency SMS Dispatch: The ESP32 formats and transmits an emergency message via the SIM800L containing GPS coordinates, altitude, a Google Maps link, and a UTC timestamp-sent to all pre-configured emergency contacts. A built-in retry mechanism reattempts transmission up to three times on failure before logging the fault to the MicroSD card.

f) Cloud Upload: Via Blynk IoT concurrently, the ESP32 uploads a complete telemetry packet-impact magnitude, tilt angle, GPS coordinates, altitude, and UTC timestamp-to the Blynk cloud platform over Wi-Fi. Location data continues streaming at regular intervals post-accident to enable live tracking via the Blynk mobile dashboard.

g) Post-Impact Voice Recording: Following SMS dispatch, the MAX9814 microphone captures 30–60 seconds of ambient post-impact audio at 8 kHz mono PCM, written to the MicroSD card in WAV format. Filenames are auto-generated from the UTC timestamp for chronological traceability, preserving audio evidence for rescue teams and forensic investigators.

h) Error Handling and System Reset: The system enters a post-alert standby state with the re-trigger lockout active. Module faults-such as GSM failure or MicroSD write errors-are logged internally with automatic soft recovery attempted. Normal monitoring resumes only after a deliberate reset via the hardware button or a remote Blynk command, returning the system to Step 1.

IV. RESULTS AND DISCUSSION

A. Detection Performance

TABLE II
System Test Results Across Simulated Accident Scenarios

Sr.	Scenario Tested	Alert Triggered	Voice Log
1	Sudden high-impact collision (simulated bike crash)	Yes	Yes
2	Sharp turn with no collision (high-speed cornering)	No	No
3	High-speed frontal impact with stationary object	Yes	Yes
4	Rough road vibration (pothole traversal at speed)	No	No
5	MPU6050 tilt beyond rollover threshold (>75°)	Yes	Yes
6	Device power-off and restart (system recovery test)	No	N/A
7	GPS signal unavailable (last-known coordinates used)	Yes	Yes
8	Minor bump at low speed (low G-force event)	No	No
9	Vehicle rollover (sustained inverted tilt detected)	Yes	Yes
10	Manual panic button press (rider-initiated alert)	Yes	Yes

The performance of the system was evaluated through ten distinct simulations designed to test the algorithmic ability to differentiate between actual accidents and high-intensity riding manoeuvres. The results, as summarized in Table II, demonstrate high reliability and a low false-alarm rate across varied conditions. In critical accident scenarios, such as sudden high-impact collisions, high-speed frontal impacts, and sustained vehicle rollovers, the system successfully triggered both alerts and voice logs. The integration of the MPU6050 was particularly effective, accurately detecting when the tilt exceeded the safety threshold of 75°.

A primary objective of this study was the suppression of false triggers using adaptive thresholding and time-based verification. By requiring the sensor data to stay beyond specific limits for a set duration, the system effectively ignored transient noise from environmental factors. Consequently, the system remained inactive during high-speed cornering, rough road vibrations from potholes, and minor low-speed bumps, ensuring that aggressive but safe riding does not initiate unnecessary emergency protocols. Furthermore, the system exhibited strong resilience during edge cases; for instance, when a GPS signal was unavailable, it defaulted to last-known coordinates to ensure an emergency dispatch was still possible. The inclusion of a manual panic button served as a reliable fail-safe, while the device power-off and restart tests confirmed that system recovery does not result in false alerts, thereby validating the overall stability and accuracy of the detection framework.

B. Communication Performance

GSM-SMS alerts were delivered in an average of 8–12 seconds, consistent with [3]. The retry mechanism successfully recovered delivery in two cases of first-attempt failure. GPS coordinate accuracy averaged 3–5 meters under open-sky conditions, enabling precise navigation to the accident site. Voice recordings averaged 960 KB per 60-second capture at 8kHz mono PCM.

V. FUTURE SCOPE

Several enhancements are identified for future development. Integration of on-device machine learning (TensorFlow Lite trained on accelerometer time-series) can replace threshold-based detection, reducing false positives from complex driving scenarios. The audio module can be upgraded to 44.1kHz sampling with wind noise suppression filters. OBD-II port integration would provide additional vehicle state data (engine RPM, airbag status) at impact, enriching the forensic log. Satellite communication (LoRaWAN or Iridium) could be added as a tertiary channel for GSM-dead zones. Miniaturization onto a custom PCB in a ruggedized enclosure would enable mass-market deployment.

VI. CONCLUSION

This paper presents a comprehensive IoT-based accident detection and emergency response system for two-wheelers that integrates impact sensing, GPS tracking, and a novel post-impact voice logging capability. By utilizing an ESP32 and MPU6050 sensor with a time-based threshold filtering algorithm, the system effectively distinguishes genuine collisions and rollovers from common riding irregularities like potholes or sharp turns. Upon detecting an accident, the device ensures communication redundancy by dispatching location-embedded alerts via GSM-SMS while simultaneously capturing ambient audio to assist in forensic investigation and victim assessment. Ultimately, this low-cost and internet-independent solution offers a practical, scalable framework designed to minimize emergency response times and enhance the safety of the global two-wheeler population.

REFERENCES

[1] World Health Organization, Global Status Report on Road Safety 2023. Geneva: WHO, 2023.

[2] M. S. Tanmay, N. R. Shivam, V. Shivanna, D. Sathya, and C. Sankaran, “Accident Detection and Emergency Support System for Scooters Based on Edge AI Technology,” in Proc. ICT4SD 2025, Lecture Notes in Networks and Systems, vol. 1652, Springer, Cham, 2026, pp. 243–252.

[3] S. Rajput, P. Sharma, and A. Verma, “IoT-Based Vehicle Accident Detection and Automated Emergency Notification System,” Int. J. Adv. Res. Comput. Commun. Eng. (IJARCCE), vol. 15, no. 1, pp. 68–75, Jan. 2026.

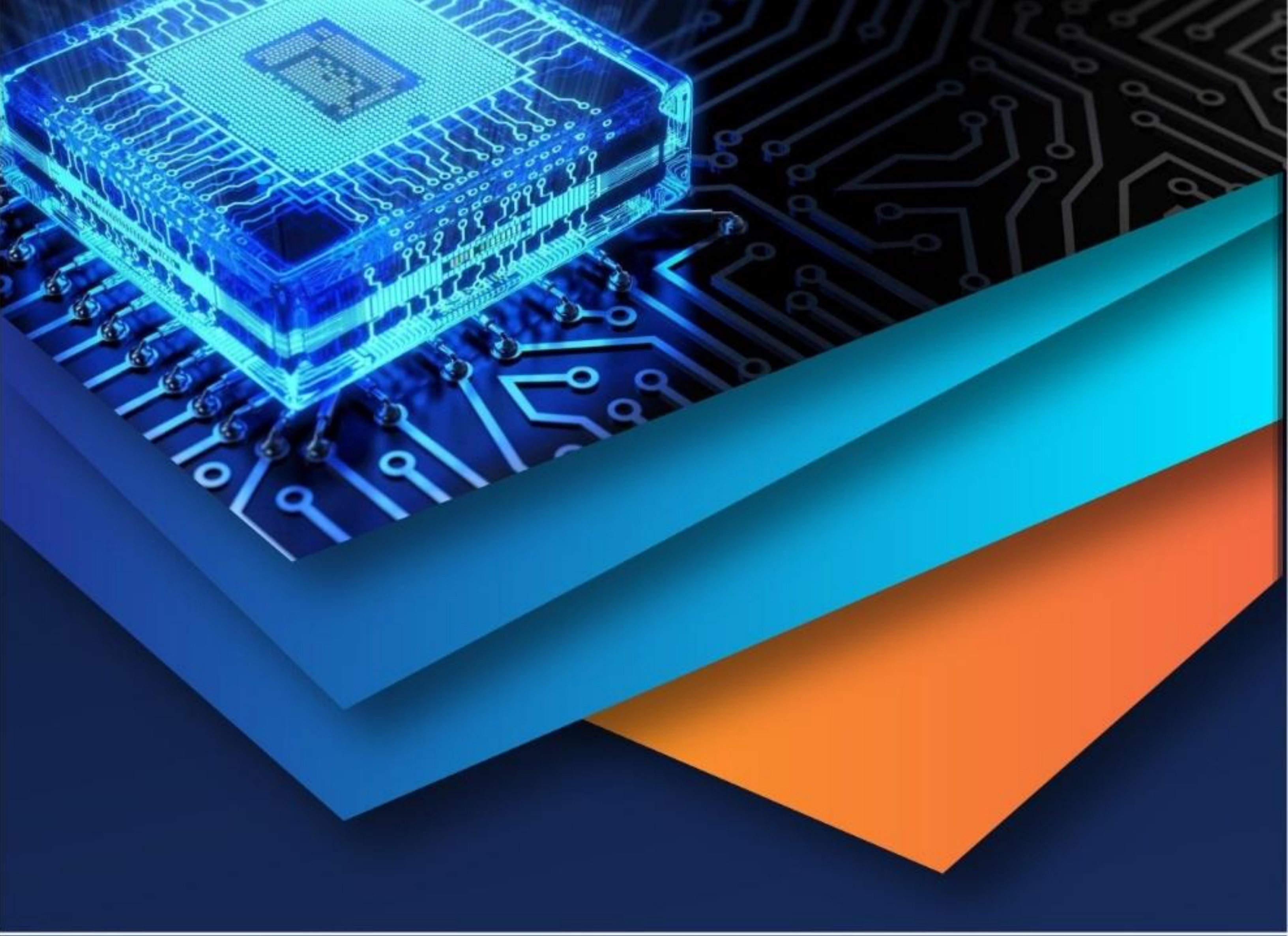
[4] P. Kadam, A. Desai, and R. Kulkarni, “Motorcycle Crash Detection and Alert System Using IoT,” in E3S Web of Conferences, vol. 391, p. 01145, EDP Sciences, 2023.

[5] R. Suryawanshi, S. Jagtap, N. Mane, S. Madhikar, Y. Munde, and N. Choudhary, “IoT-Enabled Smart Helmet: Enhancing Safety for Motorcycle Riders Through Alcohol, Drowsiness, and Helmet Detection,” in Artificial Intelligence in Internet of Things (IoT): Key Digital Trends, Lecture Notes in Networks and Systems, vol. 1072, Springer, Singapore, 2025, pp. 401–412.

[6] P. Rodegast, S. Maier, J. Kneifl, and J. Fehr, “On Using Machine Learning Algorithms for Motorcycle Collision Detection,” Discover Applied Sciences, Springer Nature, Jun. 2024.

[7] V. Krishnamurthy, S. Anand, and T. Ramesh, “An Efficient Automatic Accident Detection System Using Inertial Measurement Through Machine Learning Techniques for Powered Two Wheelers,” Expert Systems with Applications, vol. 187, p. 115954, Jan. 2022.

[8] M. Selmanaj, M. Corno, G. Osorio, and S. M. Savaresi, “Accelerometer-Based Data-Driven Hazard Detection and Classification for Motorcycles,” in Proc. IEEE Intelligent Vehicles Symposium (IV), 2014, pp. 141–146.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY