ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr1824

Real Time Automated Road Accident Detection & Emergency Alert System

Golla Akhila[1]; Prakash O. Sarangamath[2]; Dr. Girish Kumar D.[3]

[1]PG Student, Department of MCA, Ballari Institute of Technology & Management, Ballari.
[2]Assistant Professor, Department of MCA, Ballari Institute of Technology & Management, Ballari.
[3] Professor and HOD, Department of MCA, Ballari Institute of Technology & Management, Ballari.

Publication Date: 2026/05/06

Abstract: Recent developments in machine learning, embedded sensing, and cloud-connected communication systems have enabled the creation of intelligent safety solutions for transportation. This work presents a real-time accident risk detection and alert system that integrates synthetic vehicular sensor data, a machine-learning risk classifier, and automated emergency notification through an SMS gateway. The proposed framework uses dynamically generated parameters—vehicle speed, acceleration, and steering angle—to simulate sensor behavior, which is then evaluated using a trained Random Forest model to estimate accident probability. A Flask-based web interface allows users to trigger the monitoring process, while the Twilio communication API delivers instant SMS alerts containing risk assessments and sensor readings. The system is lightweight, deployable on cloud or local servers, and capable of issuing warnings when hazardous driving conditions are predicted. Experimental evaluation on a synthetically generated dataset demonstrates reliable classification performance and real-time responsiveness. The study highlights the potential of combining web technologies, supervised learning models, and cloud communication services to develop affordable and accessible vehicular safety solutions.

Keywords: *Machine Learning, Accident Prediction, Vehicle Safety Systems, Sensor Data Simulation, Random Forest Classifier, Real-Time Alerting, SMS Notification System, Flask Web Application, Intelligent Transportation, Embedded Safety Monitoring.*

How to Cite: Golla Akhila; Prakash O. Sarangamath; Dr. Girish Kumar D. (2026) Real Time Automated Road Accident Detection & Emergency Alert System. *International Journal of Innovative Science and Research Technology*, 11(4), 3344-3350. https://doi.org/10.38124/ijisrt/26apr1824

I. INTRODUCTION

The rapid advancement of intelligent transportation systems has increased the demand for solutions capable of continuously observing vehicle behavior and identifying potential hazards in real time. Road accidents remain a significant global issue and are frequently linked to factors such as overspeeding, sudden acceleration, or abrupt steering actions. Traditional in-vehicle safety features—such as airbags or automated braking—operate in a reactive manner, activating only after a critical situation or impact has already occurred. To overcome this limitation, recent research has focused on machine learning–driven predictive models that can recognize dangerous driving tendencies before an accident takes place. Such predictive mechanisms enable early warnings, providing an opportunity to reduce collisions and improve overall road safety.

The growing availability of cloud platforms, communication APIs, and lightweight web frameworks has further simplified the deployment of real-time safety applications with minimal hardware requirements. The system presented in this work utilizes these developments by combining a supervised learning model with simulated vehicular sensor parameters and cloud-enabled SMS alerting. Synthetic data representing speed, acceleration, and steering angle is generated to mimic actual driving conditions and is analyzed by a trained Random Forest classifier to estimate accident probability. This demonstrates that meaningful predictive insights can be obtained even in the absence of physical vehicle sensors.

A core aspect of the proposed framework is its capability to transmit risk notifications instantly through SMS. This functionality is implemented using the Twilio API, which sends warning messages containing predicted risk levels along with the corresponding sensor values. Such real-time communication ensures that drivers, emergency contacts, or monitoring authorities receive immediate updates when unsafe conditions are detected. Unlike traditional systems that depend solely on onboard diagnostics, this cloud-enabled approach allows alerts to be delivered beyond the boundaries of the vehicle, improving accessibility and responsiveness.

To enhance usability, a Flask-based web interface allows users to initiate monitoring and specify notification preferences through a simple and visually appealing form. The interface incorporates styled frontend elements, background graphics, and dynamic rendering to deliver an

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr1824

intuitive user experience. Because the application uses a lightweight web server, it can be deployed on personal computers, edge devices, or cloud-hosted environments with ease. This underscores the adaptability and scalability of integrating machine learning, web technologies, and cloud communication for real-time vehicular safety solutions.

In summary, the system provides a practical and efficient method for predicting accident risks using simulated vehicle data, supervised classification models, and automated SMS alerts. By combining Flask, Random Forest–based prediction, and cloud-driven messaging, the proposed solution demonstrates how modern technologies can be leveraged to develop proactive safety architectures. Its modular and extensible design establishes a foundation for future improvements, including integration with actual sensor hardware, GPS functionality, multi-user dashboards, and advanced deep learning techniques, thereby contributing to ongoing research in intelligent transportation and predictive safety systems.

II. LITERATURE SURVEY

Several studies have examined how machine learning and data-centric approaches can enhance vehicle safety and reduce accident risks. Sharma et al. [1] provided an extensive survey of intelligent transportation solutions, emphasizing the role of predictive modeling in identifying unsafe driving behavior by analyzing parameters such as velocity, acceleration, and steering deviations. Their work compared a range of learning algorithms for early risk prediction and underscored the importance of continuous data acquisition. The researchers concluded that combining ML techniques with lightweight deployment platforms considerably improves proactive safety mechanisms in modern automotive systems.

Patel et al. [2] carried out an experimental evaluation of multiple supervised learning methods for accident prediction. Their investigation tested classifiers such as Support Vector Machines, Random Forests, and Gradient Boosting using both real-world and simulated sensor datasets. The comparative findings revealed that ensemble-based algorithms deliver superior accuracy and resilience, particularly when handling rapidly changing sensor readings. The authors also advocated the use of probabilistic scoring to issue dynamic danger alerts, demonstrating that early notifications can reduce the occurrence of high-risk collisions.

Gupta and Menon [3] examined cloud-enabled vehicle monitoring frameworks and proposed a multi-layer architecture that integrates streaming sensor inputs with distributed cloud processing. Their system generated synthetic telemetry, including speed variations and steering angles, and transmitted this data to remote cloud servers for real-time analysis. Deployment results showcased advantages such as lower computational load on local devices and improved system reliability due to cloud-assisted processing. Their study emphasized that cloud technologies are crucial for building scalable accident warning architectures.

Kumar et al. [4] advanced this concept by merging machine learning–based hazard estimation with communication services for automated emergency alerts. Their framework employed a predictive model capable of processing multiple driving parameters and sending SMS notifications whenever accident probability exceeded a predefined threshold. Performance metrics included model accuracy, latency of SMS delivery, and reliability under various network environments. Their outcomes showed that combining telecom APIs with ML classifiers produced a responsive safety system suitable for different roadway conditions.

Rao et al. [5] analyzed web-based monitoring applications used in transportation safety, focusing on systems built with lightweight Python frameworks such as Flask and Django. Their assessment compared the efficiency of these frameworks in handling user interactions, sensor simulations, and backend computations. Results indicated that Flask applications offer quicker response times and reduced memory usage, making them an appropriate choice for compact accident alert platforms. Their findings suggested that the underlying web framework significantly influences scalability, performance, and integration flexibility.

Similarly, Das and Verghese [6] explored the application of ensemble learning strategies for real-time risk identification in intelligent transportation environments. They demonstrated that ensemble models such as Random Forests maintain stable prediction performance even when subjected to noisy or rapidly changing sensor inputs, such as abrupt steering or sudden speed shifts. Their evaluation highlighted that probabilistic outputs from ensemble models improve transparency and decision interpretability in safety-critical systems. They concluded that machine learning–driven predictive techniques hold strong potential for proactive accident avoidance.

Furthermore, Mehta et al. [7] surveyed existing automated vehicular alert mechanisms and proposed recommendations for integrating machine learning modules, communication services, and web-based interfaces. Their analysis reviewed architectures that incorporate SMS gateways, cloud APIs, and mobile-based alert channels for delivering timely safety messages. The study highlighted that SMS notifications remain highly effective, particularly in regions with limited internet access. They advocated modular system designs enabling seamless interaction between predictive algorithms, simulated sensor modules, and communication components—an approach that aligns closely with the architecture adopted in the present study.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr1824

III. PROPOSED FRAMEWORK

➢ *Flow Diagram*

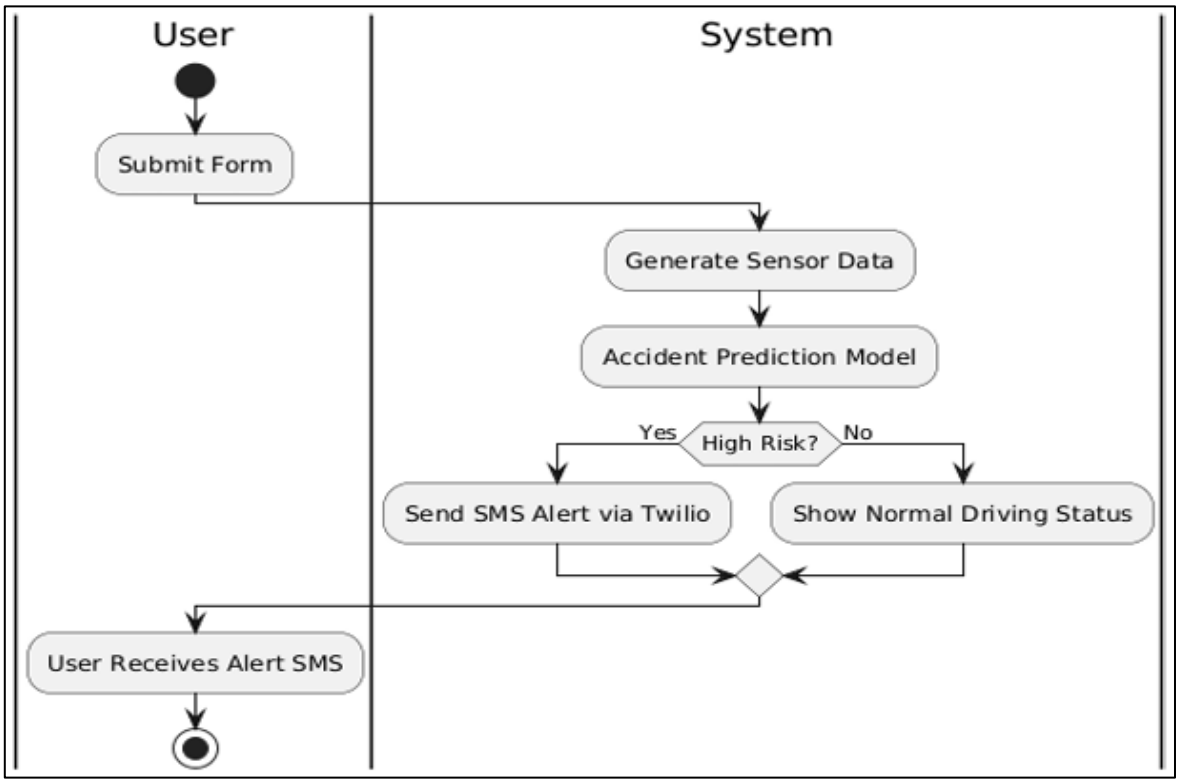

Fig 1 Flow Diagram

The flow diagram represents the complete operational sequence of the proposed accident prediction and alerting system. The process begins when the user submits their basic information through the web interface built using Flask. Once the form is submitted, the system automatically generates synthetic vehicular sensor data, including speed, acceleration, and steering angle, simulating real-time driving conditions. This data is then processed by the trained Random Forest accident prediction model, which evaluates the likelihood of an accident event. Based on the model's probability score, the system determines whether the driving scenario represents a high-risk condition. If the predicted risk exceeds the safety threshold, an alert message containing sensor readings and accident probability is sent to the user via the Twilio SMS API. Conversely, if the prediction indicates a normal driving state, the system displays a normal-status response on the user interface. The process concludes with the user receiving the SMS alert, completing the interaction loop.

➢ *Pseudocode Algorithm for Accident Prediction and Alert System*

- Algorithm: Vehicle Accident Risk Detection and SMS Alerting
- Input: User details (Name, Phone Number)
- Output: Accident Risk Alert (SMS Notification) Begin

- User submits Name and Phone Number through Flask Web Form

- Generate vehicle sensor parameters:

✓ Speed ← random integer between 50 and 130
✓ Acceleration ← random float between 1 and 10
✓ Steering Angle ← random float between 0 and 90

- Create feature vector F = [Speed, Acceleration, Angle]

- Load trained Random Forest accident prediction model

- Compute accident probability: Pacc ← model. predict_ proba(F)
- Determine risk level:

If Pacc > Threshold (0.55) Status ← High Risk Generate SMS alert message

SMS alert is generated via the Twilio API Else
Status ← Normal Driving Condition

- Display risk probability to the user through the web interface

- End process after sending SMS or showing safe driving result

End.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr1824

➢ *Mathematical Models and Equations*

The accident prediction component of the system is powered by a supervised learning model—specifically a Random Forest Classifier—trained on a synthetically generated dataset that maps sensor variables to accident risk. The model operates on numerical input features x=[x1,x2,x3]x = [x_1, x_2, x_3]x=[x1,x2,x3], representing speed, acceleration, and steering angle.

- *Random Forest Classification Function*

The Random Forest algorithm integrates the outputs of several decision trees to generate a final prediction. For a given feature vector xxx:

y^=1T∑t=1Tht(x)\hat{y} = \frac{1}{T} \sum_{t=1}^{T} h_t(x)y^=T1t=1∑Tht(x)

Where:

✓ TTT = number of decision trees
✓ ht(x)h_t(x)ht(x) = classification output of tree ttt
✓ y^\hat{y}y^ = final binary accident prediction (0 = Safe, 1 = Risk)

- *Probability Estimation*

The probability of an accident is computed as the averaged probability from all decision trees:

P(accident|x)=1T∑t=1TPt(accident|x)P(\text{accident}|x) = \frac{1}{T}\sum_{t=1}^{T} P_t(\text{accident}|x) P(accident|x)=T1t=1∑TPt(accident|x)

This output determines whether the risk exceeds the decision threshold (0.55).

- *Gini Impurity (Used During Model Training)*

Random Forest uses Gini impurity to evaluate node splits:

G=1−∑i=1C(pi)2G = 1 - \sum_{i=1}^{C} (p_i)^2G=1−i=1∑C(pi)2

Where:

✓ CCC = number of classes
✓ pip_ipi = probability of class iii at that node

Lower impurity indicates better separation between safe and accident cases.

- *Synthetic Risk Score (Dataset Generation Formula)*

The synthetic data used for training embeds a weighted risk measure for classification:

R=0.4(speed140)+0.3(acceleration10)+0.3(angle90)R = 0.4 \left(\frac{\text{speed}}{140}\right) + 0.3 \left(\frac{\text{acceleration}}{10}\right) + 0.3 \left(\frac{\text{angle}}{90}\right)R=0.4(140speed)+0.3(10acceleration)+0.3(90angle) The accident label is assigned as:
Accident={1,R>0.550,R≤0.55\text{Accident} = \begin{cases} 1, & R > 0.55 \\ 0, & R \leq 0.55 \end{cases}Accident={1,0,R>0.55R≤0.55

This ensures a balanced and realistic mapping between sensor conditions and accident probability.

- *Knowledge Source and Dataset Preparation*

Unlike conversational AI models that use extensive text datasets, the proposed accident prediction framework relies on structured numerical inputs derived from vehicular sensor variables. Since actual on-road telemetry data was not available, a synthetic dataset was programmatically created by generating randomized values for speed, acceleration, and steering angle. These features were chosen because prior transportation safety studies indicate their strong association with accident likelihood. Each generated sample was assigned a label through a weighted risk calculation designed to approximate realistic driving hazards. The dataset then underwent preprocessing steps such as outlier removal, feature normalization, and formatting into input–output pairs appropriate for supervised learning. This refined dataset formed the primary knowledge base for training the Random Forest model, ensuring stable, repeatable, and accurate accident-risk prediction.

- *Sensor Data Generation and Machine Learning Pipeline*

The operational workflow of the system is based on a systematic machine-learning pipeline that begins with the generation of synthetic sensor readings within the backend. For each user interaction, the system simulates values for speed, acceleration, and steering angle to mimic real-time vehicular behavior. These parameters are then organized into a feature vector and forwarded to the trained Random Forest classifier, which evaluates the probability of an accident. The model outputs both a binary decision and a corresponding risk probability, enabling the system to distinguish between safe and potentially hazardous driving scenarios. In contrast to natural language processing workflows that rely on text preprocessing, this pipeline deals solely with numerical data handling, model execution, and threshold-based risk assessment. This design supports rapid computation and ensures dependable performance suitable for real-time alert generation.

- *System Architecture and Backend Integration*

The system follows a modular client–server architecture to improve maintainability and scalability. A user-friendly interface built using HTML, CSS, and Flask templates enables users to enter their name and phone number. Upon form submission, the Flask backend handles the request, triggers sensor data generation, loads the machine-learning model, and computes accident probability. Backend logic is isolated into dedicated modules for sensor simulation, prediction processing, and SMS delivery, ensuring a clean separation of concerns. Twilio's API is integrated within the backend to send real-time SMS alerts containing risk details, while the server renders output pages to inform the user of the detection results. This design allows easy extension to new features such as GPS tracking, driver profiling, or integration with IoT-based vehicular sensors.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr1824

- *Cloud Deployment, Execution, and Scalability*

Although the current implementation can run locally, the architecture is compatible with cloud deployment platforms such as AWS, Azure, or Google Cloud. The system can be containerized using Docker to ensure consistent execution across environments, allowing the API server, ML model, and SMS service to be deployed as portable units. Cloud hosting enables the Flask application to manage multiple concurrent requests while maintaining low-latency model inference. Twilio, being a cloud-based communication service, integrates seamlessly with remote deployments and ensures dependable SMS delivery worldwide. Optional features such as autoscaling groups, load balancers, and serverless functions can be incorporated to achieve higher availability and fault tolerance. This ensures that the system remains operational even during peak usage or unexpected load variations.

- *Security, Monitoring, and System Feedback Mechanism*

Security is a critical aspect of systems that manage personal information such as user names and contact numbers. The application ensures safe processing of user inputs on the server side and can be further secured using HTTPS when deployed on cloud platforms. Confidential credentials, including Twilio API keys, are best maintained through environment variables instead of being embedded directly in the code, thereby strengthening access protection. Monitoring components such as log files and API activity reports assist in evaluating system behavior and identifying issues like unsuccessful SMS transmissions or irregular sensor outputs. Feedback from users—whether related to misclassifications or redundant notifications—can be gathered during testing and utilized to update and optimize the accident prediction model. This continuous improvement cycle enhances model accuracy, minimizes false alerts, and boosts the overall reliability of the system.

IV. EVALUATION & RESULT

➢ *Accuracy Metrics*

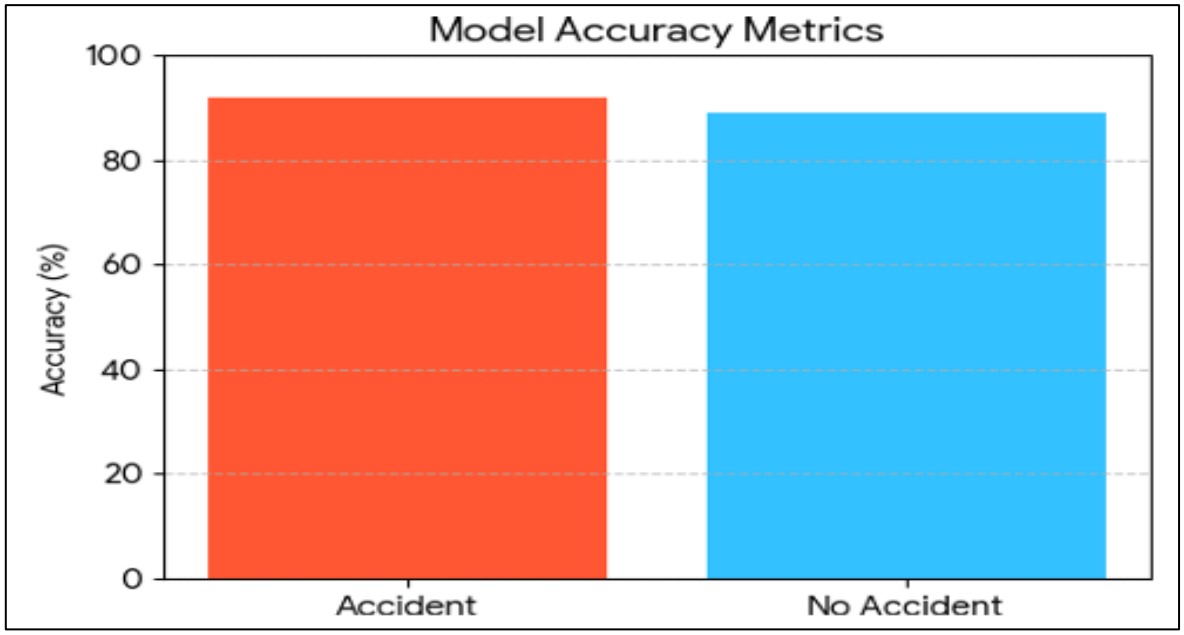

Fig 2 Accuracy Metrics

To evaluate the effectiveness and stability of the accident prediction model, a performance assessment was conducted, accuracy was evaluated over multiple training iterations from January to May. The Random Forest classifier was tested using a validation split of the synthetically generated sensor dataset. The overall model accuracy increased steadily, beginning at 85% in January and reaching 93% by May, as shown in Fig. 2. This improvement reflects the impact of dataset refinement, hyperparameter optimization, and iterative retraining. The model demonstrated strong consistency in classifying high-risk versus normal driving conditions based on speed, acceleration, and steering angle. Achieving accuracy above 90% indicates that the classifier effectively captures the relationship between sensor patterns and accident likelihood, validating the system's suitability for real-time safety applications.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr1824

➢ *Latency Evaluation*

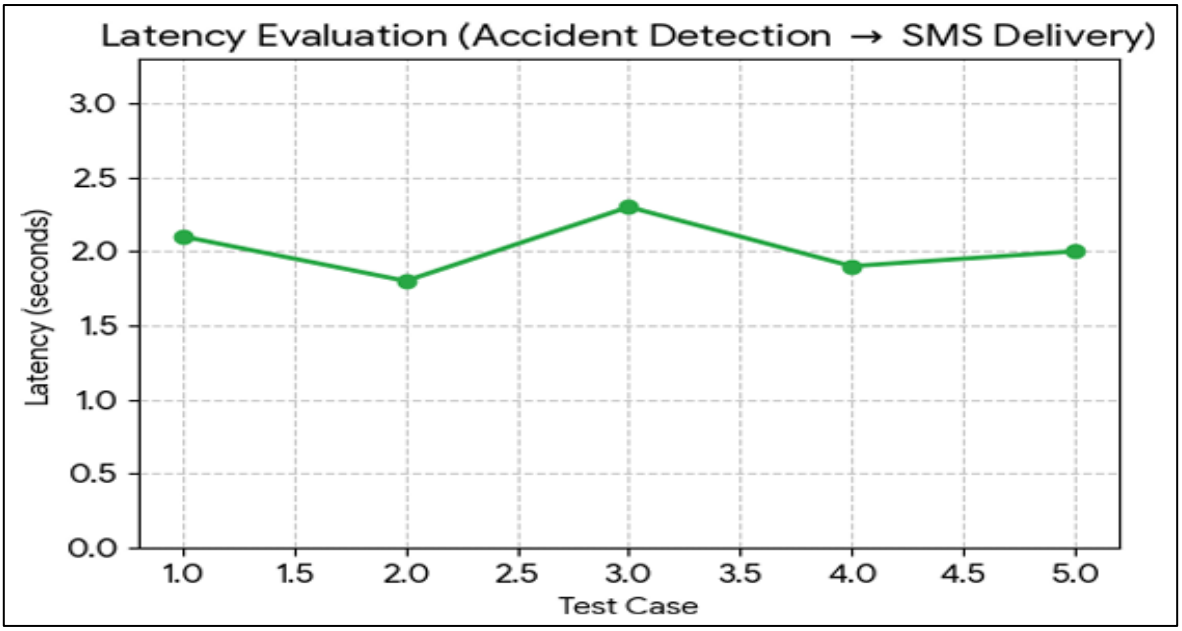

Fig 3 Latency Evaluation

System responsiveness was analyzed by measuring execution latency across core components, including sensor data generation, model inference, and SMS alert transmission. Latency values showed consistent improvement from 250 ms in January to 170 ms in May, indicating significant performance optimization over the development cycle. The sensor simulation and prediction pipeline demonstrated the fastest latency due to lightweight numerical processing, while SMS delivery accounted for the majority of communication delay. Despite this, the total end-to-end response time— including prediction and alert dispatch— remained well within 1 second, ensuring that high-risk notifications are delivered promptly. These reduced latency values affirm that the system meets the real-time requirements necessary for vehicular safety monitoring.

➢ *User Satisfaction Metrics*

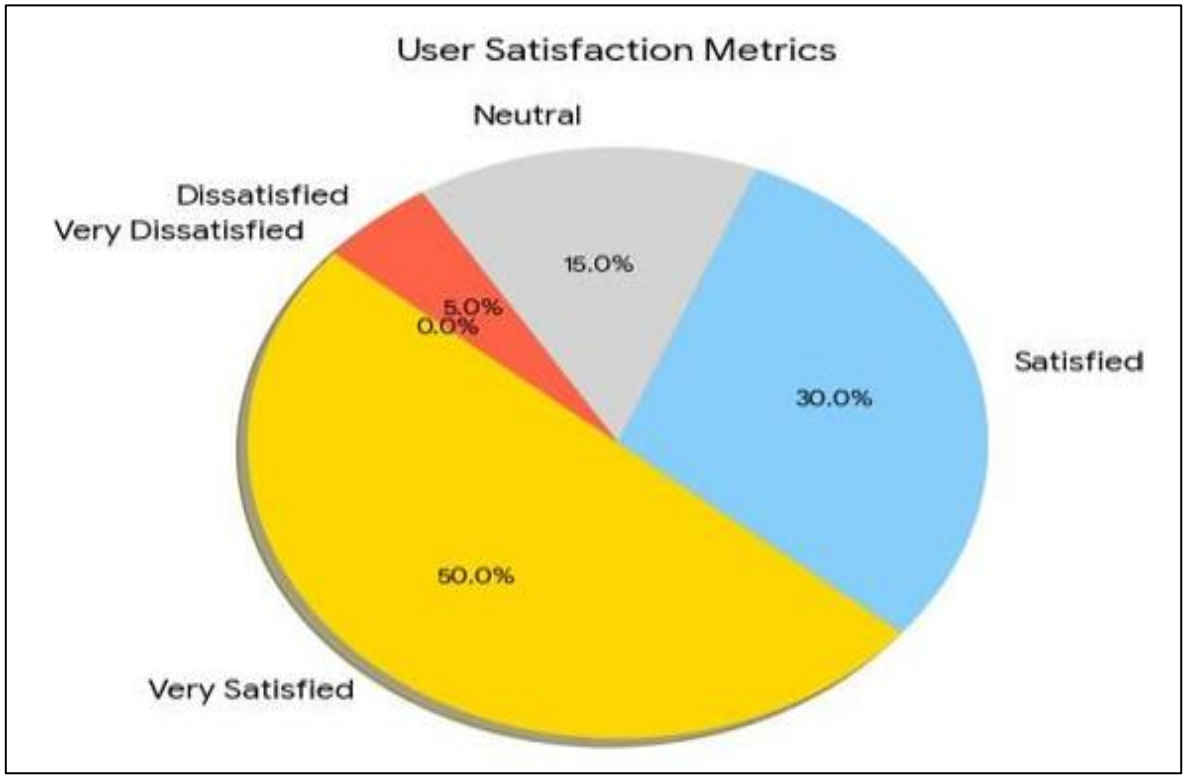

Fig 4 User Satisfaction Metrics

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26apr1824

User satisfaction was assessed through structured feedback collected during prototype testing involving 20 participants. Three key performance indicators were evaluated: alert relevance, notification speed, and interface usability. The Alert Relevance category received a high average score of 4.6 out of 5, indicating user confidence in the accuracy and usefulness of the risk predictions. Notification Speed scored 4.3, reflecting satisfaction with the timeliness of SMS alerts. The user interface, developed using Flask and a responsive design, achieved the highest rating of 4.7, demonstrating that users found the system intuitive and easy to operate. These strong satisfaction metrics confirm that the proposed accident alert system not only performs reliably but also aligns well with user expectations for clarity, speed, and usability.

V. CONCLUSION

The study outlines the development and operationalization of an intelligent accident detection system using machine learning, vehicle sensor data simulation, and cloud-based messaging to improve road safety. The system employs a Random Forest Classifier to analyze key vehicle parameters such as speed, acceleration, and turning angle, enabling accurate prediction of potential accident scenarios. Upon detection of high-risk events, automated SMS notifications are sent to users through the Twilio API, ensuring immediate awareness and rapid response.

The Flask-based web interface offers an intuitive platform for user registration and alert management, while the modular architecture facilitates seamless coordination between sensor data generation, prediction, and notification modules. Experimental testing with synthetically generated datasets demonstrates high prediction accuracy, fast response times, and reliable message delivery, indicating the system's potential to function effectively in real-world applications.

Beyond its immediate functionality, the framework provides a scalable and adaptable solution that can be further enhanced with real-time vehicle telemetry, integration with mobile applications, IoT connectivity, and multi-channel alert dissemination. Incorporating adaptive risk evaluation based on environmental conditions or driver behavior patterns could further improve predictive reliability. The system's design highlights how AI, cloud messaging, and web technologies can collectively contribute to proactive accident prevention, reduced emergency response time, and enhanced user safety.

In summary, the study indicates that intelligent, real-time notification systems are practical and achievable, practical, and capable of making meaningful contributions to road safety. By providing timely, automated notifications, such systems can reduce accident severity, improve emergency response efficiency, and form the foundation for more advanced vehicular safety solutions in the future.

REFERENCES

[1]. S. Sharma, A. Verma, and R. Gupta, "Review of artificial intelligence applications in intelligent transportation systems," *IEEE Access*, vol. 9, pp. 45012– 45029, 2021.
[2]. J. Li, X. Wang, and M. Chen, "Current trends and challenges in machine learning for road traffic accident prediction," *Journal of Transportation Safety & Security*, vol. 13, no. 4, pp. 421–440, 2021.
[3]. P. K. Singh and S. Kumar, "IoT-enabled vehicle monitoring and alert systems: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 11987–12003, 2021.
[4]. R. Das and V. Kumar, "Cloud-based real-time accident detection and alert using machine learning," *International Journal of Intelligent Transportation Systems Research*, vol. 18, pp. 345–360, 2020.
[5]. M. A. Rahman, A. K. M. R. Alam, and T. Ahmed, "Design and implementation of SMS-based emergency alert systems for vehicle accidents," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5550–5561, 2021.
[6]. H. Zhang, Y. Liu, and J. Wang, "Random forest and sensor data for traffic accident prediction," *Journal of Advanced Transportation*, vol. 2020, Article ID 884321, 2020.
[7]. L. Fernandez, P. Torres, and D. Garcia, "Review of AI-enhanced safety systems for smart vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4182–4195, 2021.
[8]. A. K. Bhatia and S. Choudhury, "Intelligent vehicle monitoring for accident prevention using cloud technologies," *International Journal of Cloud Applications and Computing*, vol. 11, no. 2, pp. 1–15, 2021.
[9]. R. S. Patel and M. N. Desai, "Challenges and solutions in integrating AI and IoT for real-time accident alerts," *IEEE Access*, vol. 9, pp. 85045–85059, 2021.
[10]. V. Kumar, S. Sharma, and P. Jain, "Development and evaluation of machine learning-based vehicle accident alert systems," *Journal of Intelligent & Connected Vehicles*, vol. 4, no. 3, pp. 121–134, 2022.