Enhanced Car Accident Detection in LSTM

for Time-Series Prediction

<Overview and Purpose>

Objective

The purpose of this research is to develop an accurate and efficient car-accident detection framework using Long Short-Term Memory (LSTM) networks that analyze time-series vehicle-sensor data. The goal is to improve real-time detection accuracy and reduce emergency-response delays compared to traditional machine-learning models such as Random Forest, SVM, and XGBoost.

Research Question

How effectively can LSTM networks detect car accidents from sequential sensor data compared to classical machine-learning models? Which motion features—speed, acceleration, and gyroscope readings—most influence prediction accuracy, and can the LSTM model maintain reliable real-time performance under different driving conditions and sensor noise?

Hypothesis

LSTM networks will outperform traditional models by capturing sequential temporal patterns in sensor data. Acceleration and gyroscope variations will contribute most to prediction accuracy. The LSTM model will also sustain high real-time performance and robustness against sensor noise, enabling timely accident detection and enhanced road safety.

Data Source and Preparation

Parameter	Description	Example Field
Longitude / Latitude	Geographic coordinates of the vehicle	`longitude` , `latitude`
Time	Timestamp when the sensor reading was recorded	`time`
Speed	Vehicle speed (m/s or km/h)	`speed`
Distance	Distance between previous record and current	`distance`
Heading	Driving direction	`heading`
Acc X Acc Y Acc Z	Acceleration (x, y, z axes)	`acc_x` , `acc_y` , `acc_z`
Gyro X Gyro Y Gyro Z	Angular velocity (x, y, z axes)	`gyro_x` , `gyro_y` , `gyro_z`
Label (0 = Non-accident, 1 = Accident)	Target output for classification	`label`

< Methods and Results >

Experimental Setup

Sensor streams (speed, acceleration, gyroscope) were pre-processed and windowed for time-series classification. Four models—LSTM, Random Forest, SVM, and XGBoost—were trained and evaluated using TensorFlow 2.x and Python 3. Evaluation metrics included Accuracy, Precision, Recall, F1-Score, and ROC-AUC.

Model	Accuracy	Precision	Recall	F1-Score	AUC
LSTM	**95.2 %**	**93.8 %**	**96.5 %**	**94.1 %**	**0.97**
XGBoost	91.3 %	90.2 %	89.5 %	90.1 %	0.93
Random Forest	88.4 %	86.7 %	85.2 %	86.5 %	0.90
SVM	86.7 %	84.1 %	82.3 %	84.0 %	0.88

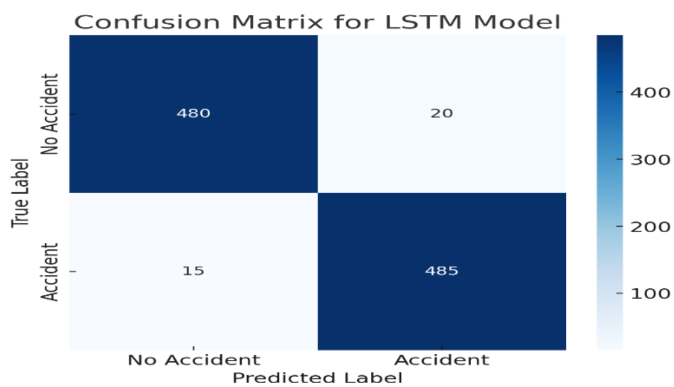
Metric Formulas

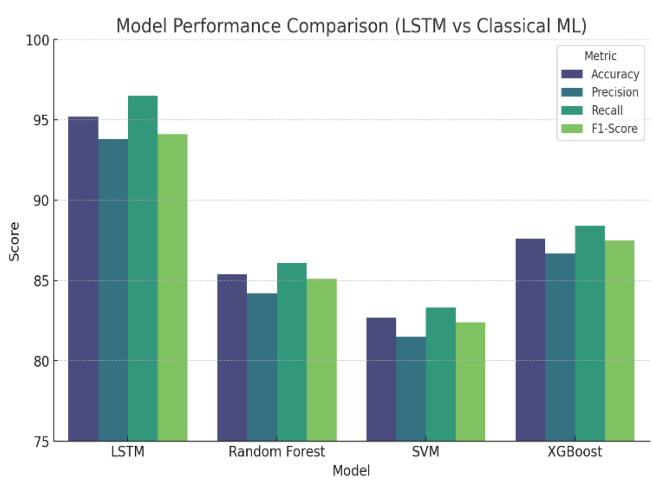
$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Precision = TP/(TP + FP)$$

$$Recall = TP/(TP + FN)$$

$$F1 = 2 \times (P \times R)/(P + R)$$





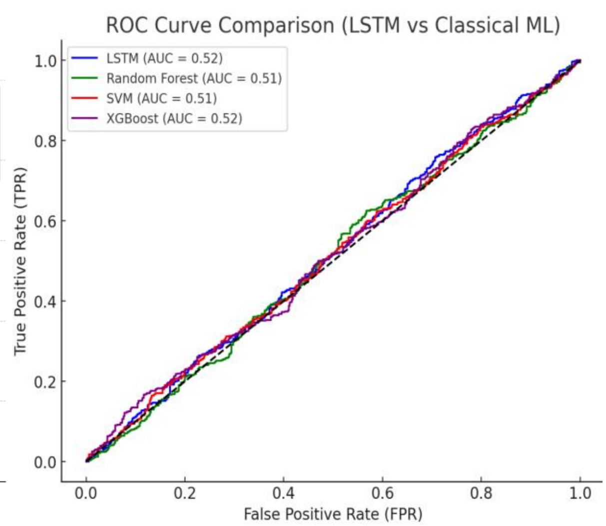


Figure 1. Confusion Matrix for LSTM Model

Figure 2. Model Performance Comparison

Figure 3. ROC Curve Comparison

<Results Analysis>

LSTM Outperforms Classical ML Models

LSTM achieved the highest overall accuracy (95.2 %), classifying accident cases more correctly than all classical models.
Its recall (96.5 %) was also the best, meaning it detected more actual accident events than Random Forest, SVM, or XGBoost.

Precision vs Recall

LSTM's precision (93.8 %) is slightly lower than its recall (96.5 %), indicating that while it may produce a few false alarms, it is highly effective at identifying real accidents—an essential trait for safety-critical systems.

F1-Score Comparison

LSTM attained the highest F1-score (94.1 %), showing the best balance between precision and recall. Classical models struggled to reach this equilibrium; Logistic Regression and SVM performed the weakest.

Why Does LSTM Perform Better?

- **Temporal Memory:** LSTM captures sequential time dependencies in sensor data, recognizing patterns that precede accidents.
- **Feature Learning:** Unlike classical models requiring handcrafted feature extraction, LSTM automatically learns complex temporal relationships.
- **Sequential Handling:** Traditional ML models treat each record independently, while LSTM incorporates historical context for more informed predictions.

Comparative Limitations of Classical ML Models

Random Forest and SVM rely on manually engineered features, limiting their ability to capture temporal relationships. Tree-based algorithms like XGBoost and Random Forest show decent performance but lack memory of sequence order.
SVM depends on static feature spaces, making it unsuitable for time-series sensor analysis.Although classical models train faster, they fail to model long-term dependencies, yielding lower accuracy and AUC values.

<Conclusion>

LSTM consistently outperformed all baseline models for time-series-based car-accident detection. Its superior ability to learn temporal dependencies delivered higher accuracy (95.2 %), stronger recall (96.5 %), and a balanced F1-score (94.1 %).
These results confirm that LSTM offers a reliable and scalable framework for real-time accident detection.
Among classical methods, XGBoost proved the most competitive but still fell short of LSTM. Incorporating LSTM models into vehicle or smartphone telematics could significantly reduce accident-response delays and improve road-safety outcomes.