Algorithmic Protocols for Smartphone-Based Telematics in West Africa: A Technical Analysis of Unsafe Driving Detection

1. Introduction: The Telematics Paradigm Shift in the West African Context

The global discourse on road safety has increasingly turned its focus to the developing world, where the disparity between vehicle ownership and road fatalities is starkest. Africa, hosting approximately 4% of the world's vehicle fleet, accounts for a disproportionate 24% of global road traffic fatalities.[1] This statistical incongruity is nowhere more visible than in the bustling economic corridors of West Africa—from the chaotic, sprawling interchanges of Lagos to the vibrant, congested arterials of Accra and Abidjan. In these environments, road safety is not merely a matter of infrastructure engineering but a complex interplay of vehicular mechanics, intense socio-economic pressures, and a driving culture forged in the crucible of fierce competition.

Traditionally, the monitoring of driving behavior has been the domain of dedicated telematics hardware—proprietary "black boxes" hardwired into the vehicle's Controller Area Network (CAN) bus. While effective, these systems present significant barriers to entry in the West African context: high installation costs, logistical difficulties in maintenance, and a lack of scalability across the fragmented, informal transport sector dominated by paratransit operators. The ubiquity of the Android smartphone, however, offers a disruptive alternative. Modern mobile devices are equipped with Micro-Electro-Mechanical Systems (MEMS) sensors of increasing fidelity, capable of capturing the kinematic signatures of unsafe driving behaviors with a granularity that rivals dedicated hardware.

This report presents a rigorous, academically grounded analysis of rule-based algorithms designed to detect specific unsafe driving behaviors—Phone Handling, Fatigue, Rough Road Speeding, Crash Detection, Geofence/Time Violations, and Aggressive Stop-and-Go—using the standard Android sensor stack. Unlike generic telematics solutions designed for the paved predictability of Western Europe or North America, these protocols are calibrated for the specific realities of the West African road network. We analyze the physics of the *Trotro* in Ghana, the *Car Rapide* in Senegal, and the *Gbaka* in Côte d'Ivoire, proposing algorithmic adjustments that distinguish between the vibration of a pothole-ridden laterite road and the jerk of a dangerous maneuver. By synthesizing data on Android sensor accuracy, MEMS noise characteristics, and regional driving sociology, we establish a comprehensive framework for smartphone-based driver profiling in the region.

2. Android Sensor Metrology and Signal Processing

The foundation of any robust telematics algorithm is a precise understanding of the instrumentation. In the context of a smartphone, the "instrument" is a consumer-grade device prioritizing battery life and screen orientation over inertial navigation. Developing valid rule-based algorithms requires a deep appreciation of the Android sensor framework's architecture, the stochastic noise characteristics of MEMS components, and the mathematical transformations required to align sensor data with the vehicle's frame of reference.

2.1 The Android Sensor Architecture

The Android platform abstracts physical sensors into a standardized API, providing raw data streams crucial for behavioral analysis. For telematics applications, we primarily rely on the motion and position sensor categories.

2.1.1 The Accelerometer (TYPE_ACCELEROMETER)

The accelerometer is the workhorse of driving behavior analysis. It measures the acceleration force applied to the device on all three physical axes (x, y, and z), including the force of gravity.[2] The output is measured in meters per second squared (m/s^2).

- **Relevance:** Used for detecting braking, acceleration, cornering, and impact events.
- **Limitations:** It cannot distinguish between gravity and vehicle acceleration on its own. A phone resting on a dashboard will report $9.81 m/s^2$ on the axis perpendicular to the ground. Algorithms must effectively subtract this gravity component to isolate the vehicle's kinematic forces.

2.1.2 The Gyroscope (TYPE_GYROSCOPE)

The gyroscope measures the rate of rotation around the device's x, y, and z axes, outputting data in radians per second (rad/s).[2]

- **Relevance:** Critical for detecting turns, lane changes, and, most importantly, phone handling. A vehicle turning generates a smooth, low-frequency rotation on the yaw axis, whereas a phone being manipulated by a driver generates high-frequency, multi-axis rotational noise.
- **Drift:** Gyroscopes suffer from bias instability, meaning the "zero" point drifts over time due to temperature changes and inherent sensor imperfections.

2.1.3 The Linear Acceleration Sensor (TYPE_LINEAR_ACCELERATION)

This is a software-fused sensor provided by the Android API. It conceptually isolates the acceleration of the device *excluding* the force of gravity.[3]

- **Mechanism:** The system typically fuses data from the accelerometer and the gyroscope (and sometimes the magnetometer) to determine the gravity vector's orientation. It then subtracts this vector from the raw accelerometer reading.
- **Utility:** For identifying aggressive braking or acceleration, this sensor is superior to the raw accelerometer as it removes the static offset of earth's gravity, theoretically leaving only the vehicle's change in velocity. However, it relies on the accuracy of the underlying sensor fusion algorithm.

2.1.4 The Rotation Vector Sensor (TYPE_ROTATION_VECTOR)

This fused sensor provides the orientation of the device relative to the Earth's coordinate system.[3] It is essential for the "reorientation" process—mathematically aligning the phone's axes with the vehicle's axes. Without this, a "braking" event (longitudinal deceleration) might register as a "cornering" event (lateral acceleration) if the phone is essentially rotated 90 degrees in the cup holder.

2.2 MEMS Sensor Characteristics and Noise Analysis

A critical failure mode in smartphone telematics is the assumption that sensor data is noise-free. In reality, MEMS sensors exhibit stochastic errors that must be filtered. Understanding these errors via Allan Variance analysis is necessary to set appropriate thresholds for event detection.

2.2.1 Noise Density and Random Walk

Noise density characterizes the random fluctuations in sensor output. For accelerometers, this is often expressed in $\mu g/\sqrt{Hz}$ or velocity random walk ($m/s/\sqrt{hr}$).

- **Standard Values:** Consumer-grade accelerometers in smartphones typically have noise densities in the range of $100-300 \mu g/\sqrt{Hz}$.[4]
- **Impact on Algorithms:** If an algorithm integrates acceleration to calculate velocity (Delta-V) during a crash, this white noise accumulates as a "random walk," causing the velocity estimate to drift. Over a short window (e.g., 100ms for a crash pulse), this drift is negligible. Over minutes, it renders position estimation impossible without GPS.

2.2.2 Bias Instability

Bias instability represents the wander of the sensor's bias over time (1/f flicker noise).

- **Experimental Data:** Recent studies comparing smartphone sensors demonstrate significant variance. For example, the **Samsung SM-A536V** has been measured to have an accelerometer bias instability of approximately $0.0013 m/s^2$, whereas the **Google Pixel 7 Pro** demonstrates superior performance with a bias instability of $0.0002 m/s^2$.[5]
- **Regional Implication:** In West Africa, where ambient temperatures can be high, thermal drift becomes a factor. A sensor calibrated in an air-conditioned office in Accra may drift

significantly when the device heats up on the dashboard of a *Trotro* in midday traffic. Algorithms must employ continuous bias estimation (e.g., detecting when the vehicle is stopped to recalibrate the zero-g offset).

2.2.3 Sampling Rate Limitations

For detecting high-frequency events like crashes or pothole impacts, the sampling rate is paramount.

- **Standard Limits:** The registerListener() method in Android generally caps standard event monitoring at 200Hz.[2]
- **Requirement:** Crash detection requires capturing the peak G-force, which may occur within a 10-20ms window. A sampling rate of 100Hz (10ms period) is the minimum viable frequency for reliable crash forensics. For general driving behavior (braking, turning), 50Hz is sufficient.

2.3 Coordinate System Transformation

Smartphones are rarely aligned with the vehicle's chassis. A phone may be in a pocket, a mount, or sliding on a seat. To analyze "braking" (longitudinal negative acceleration), we must mathematically rotate the phone's raw readings into the vehicle's reference frame.

The transformation is defined as:

$$\mathbf{a}_{vehicle} = \mathbf{R}_{align} \cdot \mathbf{a}_{phone}$$

Where $\mathbf{a}_{vehicle}$ is the acceleration vector in the vehicle frame, $\mathbf{a}_{phone}$ is the raw sensor vector, and $\mathbf{R}_{align}$ is the rotation matrix.

Dynamic Reorientation Algorithm:

1. **Detect Gravity:** Identify the "Down" vector using the TYPE_GRAVITY sensor or by low-pass filtering the accelerometer data over a long window (gravity is the constant force). This establishes the vertical (Z) axis.[6]
2. **Detect Forward Motion:** Identify the "Forward" vector by monitoring GPS heading changes or analyzing the long-term average of linear acceleration (acceleration usually occurs in the forward direction, braking in the backward direction).
3. **Compute Matrix:** Construct a rotation matrix that maps the phone's "Down" to the vehicle's Z-axis and the phone's "Forward" to the vehicle's Y-axis.

3. Algorithm I: Phone Handling Detection

Distracted driving is a primary catalyst for road accidents. In the dense, unpredictable traffic of West African cities, the cognitive load of driving is high. Using a phone manually—whether texting, calling, or browsing—significantly degrades driver performance. The challenge lies in distinguishing a phone moving in a hand from a phone moving in a pocket or vibrating in a mount.

3.1 The Physics of Handling vs. Vehicle Motion

A phone fixed to a vehicle (mounted) moves as a rigid body with the vehicle. It experiences:

- **Low Gyroscopic Variance:** Rotations are limited to the vehicle's yaw (turning) and slight pitch/roll from road gradients.
- **Correlated Acceleration:** Acceleration correlates strongly with GPS speed changes.

A phone being handled experiences:

- **Micro-Rotations:** The human hand is not perfectly steady. Handling introduces high-frequency rotational noise across all three axes.
- **Independent Orientation Change:** The screen angle relative to gravity changes rapidly as the user tilts the device to view it.

3.2 Rule-Based Detection Protocol

The proposed algorithm utilizes a "Variance of Magnitude" approach on the gyroscope data.

Step 1: Vehicle Motion Gate
The algorithm only activates when the vehicle is moving to avoid flagging phone use while parked (which is generally legal and safe).

$$Speed_{GPS} > Threshold_{motion} \quad (\text{e.g., } 10 \, km/h)$$
Step 2: Gyroscopic Energy Calculation
We calculate the magnitude of the angular velocity vector $\boldsymbol{\omega}$ at each timestamp t:

$$|\boldsymbol{\omega}_t| = \sqrt{\omega_{x,t}^2 + \omega_{y,t}^2 + \omega_{z,t}^2}$$
Step 3: Variance Analysis (The "Jitter" Test)
We calculate the variance (σ^2) of $|\boldsymbol{\omega}|$ over a sliding window (e.g., $W = 2.5$ seconds).

$$\sigma^2_{gyro} = \frac{1}{N} \sum_{i=1}^{N} (|\boldsymbol{\omega}_i| - \mu_{\omega})^2$$

Thresholding:

- **Mounted:** $\sigma^2_{gyro} < 0.05 \, (rad/s)^2$.

- Handling: $\sigma^2_{gyro} > 0.15 \, (rad/s)^2$.
  Research indicates that hand movements introduce significant "jitter" that vehicle suspension dampens. A threshold of approximately $0.5 \, rad/s$ for absolute rotation rate or a high variance is a strong indicator of handling.7

Step 4: The "Pocket" Filter
A phone in a pocket can swing, mimicking handling.

- **Proximity Sensor Gate:** If the proximity sensor returns "NEAR" (0 cm) continuously, the phone is likely in a pocket or bag. The handling alert is suppressed.
- **Screen State Gate:** If accessible (dependent on Android permissions), checking isInteractive() verifies if the screen is ON.
  - *Rule:* IF ($Speed > 10$) AND ($GyroVariance > Threshold$) AND ($Screen == ON$) AND ($Proximity == FAR$) THEN **DETECT_HANDLING**.

3.3 West African Context: The "Mate" Factor

In Ghana's *Trotro* system, a conductor (the "mate") often sits next to the driver. In a crowded front seat, the mate might handle a phone, or the driver might pass a phone to the mate.

- **differentiation Challenge:** A sensor-only approach cannot easily distinguish *who* is holding the phone.
- **Centripetal Acceleration Solution:** By comparing the lateral acceleration measured by the phone (a_{lat}) with the theoretical centripetal acceleration derived from GPS speed and turn radius (v^2/r), it is theoretically possible to estimate the phone's distance from the center of rotation.[8] However, given the variability in *Trotro* seating layouts, this is often computationally expensive and prone to error.
- **Operational Recommendation:** For commercial telematics in this region, the system should assume the phone belongs to the driver if the app is logged into a "Driver" profile, placing the onus of compliance on the registered user.

4. Algorithm II: Fatigue Detection

Fatigue management is critical for inter-city transport in West Africa, where journeys such as Lagos to Abuja or Accra to Kumasi involve long hours on challenging roads. Unlike camera-based systems that monitor eye closure (PERCLOS), sensor-based fatigue detection relies on detecting the degradation of motor skills, specifically steering control.

4.1 Circadian Rhythms and Time-on-Task

Human alertness follows a circadian rhythm, with dips in performance typically occurring between **02:00–06:00** and **14:00–16:00**.[9] Additionally, "time-on-task" fatigue sets in after prolonged continuous driving.

Rule-Based Logic:

1. **Time-of-Day Weighting:** Events occurring during circadian dips are weighted with a higher "Fatigue Probability."
2. **Hours of Service (HOS) Violation:** Based on international standards (e.g., FMCSA) adapted for regional feasibility:
   - *Rule:* IF ($ContinuousDriving > 4.5 \, hours$) THEN **FATIGUE_WARNING_LEVEL_1**.
   - *Rule:* IF ($DailyDriving > 10 \, hours$) THEN **FATIGUE_WARNING_LEVEL_2**.
   - *West African Reality:* Many *Trotro* drivers work 14-hour shifts to meet "sales" targets.[10] While commercially common, this is biologically unsafe. The algorithm must log these violations for fleet managers to adjust shift patterns.

4.2 Steering Reversal Rate (SRR) Analysis

Fatigued drivers exhibit a specific steering pattern: they drift slowly out of the lane (lapse in attention) and then jerk the wheel to correct (over-correction). Alert drivers, by contrast, make frequent, small, smooth micro-corrections to maintain lane position.

The Algorithm:

1. **Isolate Yaw Rate:** Use the reoriented gyroscope Z-axis data (ω_z).
2. **Filter Noise:** Apply a low-pass filter (e.g., 2Hz cutoff) to remove road vibration noise.
3. Count Zero Crossings (SRR): A "reversal" occurs when the rotation rate changes sign (from left-turn to right-turn or vice versa).

   $$N_{reversals} = \text{Count}(\text{sign}(\omega_{z,t}) \neq \text{sign}(\omega_{z,t-1}))$$
4. **Thresholding:**
   - *Fatigue Signature:* Low count of reversals (drifting) coupled with high magnitude of reversals (jerky correction).
   - *Alert Signature:* High count of reversals (active tracking) with low magnitude.
   - *Standard:* A decrease in SRR below $0.5 Hz$ with large angular corrections often signals drowsiness.[9]

4.3 Environmental Noise: The Pothole Problem

In West Africa, drivers frequently swerve to avoid potholes or motorcycle taxis (*Okadas*). This active avoidance behavior generates a *high* SRR and high magnitude, which a standard algorithm might misinterpret as "Alert/Aggressive" rather than "Fatigued."

- **Contextual Filter:** The fatigue algorithm must be **suppressed** or **recalibrated** when the "Rough Road" algorithm (see Section 5) detects high Z-axis vibration. Fatigue detection via steering is only reliable on relatively smooth highway segments (e.g., the Accra-Tema Motorway).

5. Algorithm III: Rough Road Speeding and

Infrastructure Anomaly Detection

West African road infrastructure is heterogeneous. A driver might transition from a newly paved asphalt highway to a degraded laterite feeder road within minutes. Driving at highway speeds on rough roads causes severe vehicle damage and loss of control.

5.1 Differentiating Potholes from Speed Bumps

Both potholes and speed bumps create vertical acceleration spikes (Z-axis), but their temporal signatures differ.

- **Speed Bump:** The vehicle rises, then falls. This creates a positive G-force spike followed by a negative one (or vice versa depending on sensor orientation), typically lasting **100ms to 300ms**.[11]
- **Pothole:** A sharp wheel drop followed by an impact. This creates a higher frequency, shorter duration shock (**< 50ms**) with a distinct "double tap" signature (front wheel then rear wheel).[12]

Algorithm (Z-THRESH and Windowing):

1. **Vertical Isolation:** $A_z' = A_z - g$.
2. **Event Trigger:** IF $|A_z'| > Threshold_{trigger}$ (e.g., $1.5g$).
3. **Classification Logic:**
   - Measure the **Pulse Width** (T_{width}) of the event.
   - IF ($T_{width} > 100ms$) AND (Signature == Up-Down) THEN **SPEED_BUMP**.
   - IF ($T_{width} < 60ms$) AND (Signature == Down-Up) THEN **POTHOLE**.

5.2 Road Quality Indexing (RQI)

Rather than detecting individual anomalies, calculating a continuous metric of road roughness allows for "Rough Road Speeding" detection.
We utilize the Root Mean Square (RMS) of the vertical acceleration over a 1-second window.

$$RQI = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (A_{z,i} - g)^2}$$

Rule-Based Alert:

- **Rule:** IF ($RQI > Threshold_{rough}$) AND ($Speed > Threshold_{speed}$) THEN **ROUGH_ROAD_SPEEDING**.
- *Regional Thresholds:* In Ghana, driving > 80 km/h on a road with an RQI corresponding to a graded dirt road is a high-risk behavior. This algorithm helps fleet managers identify when drivers are abusing vehicles on bad routes.

6. Algorithm IV: Crash Detection

Crash detection is the most critical safety feature. It requires distinguishing a vehicular collision from non-crash high-G events, most notably dropping the phone.

6.1 Physics of Impact vs. Drop

- **Phone Drop:**
  - *Phase 1:* Freefall (0g) for ~400-600ms (dropping from hand height).
  - *Phase 2:* Impact. Extremely high peak acceleration ($>20g$, often clipping the sensor) for a very short duration (**< 10ms**).[14]
  - *Velocity Change:* The *phone's* velocity changes, but the *GPS speed* of the vehicle remains constant.
- **Vehicle Crash:**
  - *Phase 1:* Pre-impact braking (optional).
  - *Phase 2:* Impact. High acceleration ($2g - 50g$) sustained for a longer duration (**60ms - 150ms**) as the vehicle crumples.
  - *Velocity Change:* Massive change in GPS speed (Delta-V) and integrated accelerometer velocity.

6.2 The Delta-V Integration Protocol

The most reliable metric for crash severity is Delta-V (ΔV), the change in velocity during the impact.

Step 1: High-G Trigger
The system continuously monitors the total acceleration magnitude.

$$|A| = \sqrt{x^2 + y^2 + z^2}$$

- **Trigger:** IF $|A| > 4g$ THEN initiate Crash Analysis Window.[15]

Step 2: Pulse Duration and Integration
Upon trigger, the system integrates the linear acceleration over a window (e.g., 120ms).

$$\Delta V_{accel} = \int_{t_0}^{t_0 + 120ms} \mathbf{a}(t) \, dt$$

- **Phone Drop Rejection:** A drop has a massive peak $a(t)$ but a tiny Δt (pulse width). The integral ΔV_{accel} will be relatively small compared to a crash where moderate Gs are sustained for 100ms.
- **Crash Confirmation Threshold:** IF $\Delta V_{accel} > 7 \, km/h$ (approx $1.94 \, m/s$) THEN **POTENTIAL_CRASH**.[16]

Step 3: GPS Validation (The "Moving Car" Test)
A phone dropped in a moving car is a false positive risk. We check the GPS speed 10-30 seconds after the trigger.

- **Rule:** IF ($Speed_{t+30s} < 5 \, km/h$) THEN **CONFIRMED_CRASH**.[17]
    - If the vehicle is still traveling at 50 km/h after the "impact," it was not a disabling crash; it was likely a phone drop or a severe pothole strike.

Step 4: Regional Calibration
In West Africa, "bumper-to-bumper" traffic often results in minor taps.
- *Severity Triage:*
    - $\Delta V > 25 \, km/h$: Major Crash (Deployment of emergency services).
    - $8 \, km/h < \Delta V < 25 \, km/h$: Minor Incident (Log for insurance claims).

7. Algorithm V: Aggressive Stop-and-Go (Longitudinal Dynamics)

Aggressive driving behavior—specifically harsh braking and rapid acceleration—is a leading indicator of crash risk and fuel inefficiency. In the competitive environment of *Gbakas* and *Trotros*, drivers often accelerate aggressively to cut off peers and brake hard to pick up passengers.

7.1 Jerk Analysis: The Smoothness Metric

Acceleration alone tells part of the story. The rate of change of acceleration, known as Jerk (j), indicates the "suddenness" of the maneuver. High jerk values correlate with driver anger, panic, or aggression.

$$j = \frac{da}{dt} \approx \frac{a_t - a_{t-1}}{\Delta t}$$

- **Rule:** IF $|j| > 10 \, m/s^3$ THEN **AGGRESSIVE_MANEUVER**.[18]
    - This metric distinguishes between a firm, controlled stop (low jerk, high deceleration) and a panic stop or erratic driving (high jerk).

7.2 Threshold Comparisons: Developed vs. West African Context

Standard telematics thresholds derived from US/EU data often produce excessive false positives in West Africa due to the necessity of aggressive driving to navigate chaotic traffic.

Parameter	Standard US/EU Threshold	Proposed West African Threshold	Rationale
Harsh Braking	$-0.25g$ to $-0.3g$	**$-0.4g$ to $-0.5g$**	Traffic density in Lagos/Accra requires harder

			braking as a norm. $-0.3g$ is often just "stopping in traffic".[19]
Harsh Accel	$+0.25g$ to $+0.3g$	**$+0.35g$**	*Trotro* engines often lack the power to achieve high Gs, but high *jerk* at low Gs should be flagged.
Harsh Cornering	$0.3g$ to $0.4g$	**$0.45g$**	Needed to account for sudden swerves to avoid potholes/Okadas (motorcycles).

Algorithm Implementation:
Using the reoriented Linear Acceleration Sensor (y-axis for longitudinal, x-axis for lateral):

- **Braking Event:** IF ($A_y < -0.45g$) AND ($Speed > 15 \, km/h$) THEN **HARSH_BRAKE**.
- **Acceleration Event:** IF ($A_y > 0.35g$) THEN **HARSH_ACCEL**.
- **Cornering Event:** IF ($|A_x| > 0.45g$) AND ($Speed > 20 \, km/h$) THEN **HARSH_TURN**.

The inclusion of a speed gate (e.g., $>15 km/h$) is vital to prevent triggering harsh braking events during low-speed maneuvering in parking lots or traffic jams.

8. Algorithm VI: Geofence and Time Violations

Geofencing allows fleet managers to enforce route compliance and restrict vehicles from high-risk areas.

8.1 Ray Casting Algorithm for Geofence Polygons

Defining a geofence as a simple radius is often insufficient for complex urban environments. We use polygonal geofences. To determine if a vehicle is inside a polygon, the **Ray Casting** (or Crossing Number) algorithm is standard.

- **Logic:** Cast a ray from the vehicle's point to infinity. Count the intersections with polygon edges.
  - **Odd** number of intersections = **INSIDE**.
  - **Even** number of intersections = **OUTSIDE**.[20]

8.2 Handling GPS Drift (Urban Canyons)

In cities like Lagos, tall buildings can cause GPS multipath errors, causing the reported location to "jump" hundreds of meters, potentially triggering a false geofence exit.

- **Buffer Zone (Tolerance):** Do not trigger an alert immediately upon crossing the boundary.
- **Time-Based Hysteresis:**
  - *Rule:* A violation is valid ONLY IF the vehicle remains outside the boundary for > 30 seconds.
  - *Rule:* The reported position must be outside the boundary by a margin greater than the GPS accuracy radius (e.g., if GPS accuracy is 20m, the vehicle must be 25m outside).

8.3 Time-Based Restrictions (Curfews)

Due to security concerns (highway robbery) or municipal regulations (restrictions on heavy trucks in daytime), time violations are critical.

- **Rule:** IF ($Time \in$) AND ($Speed > 0$) AND ($Ignition == ON$) THEN **CURFEW_VIOLATION**.

9. Conclusion

The development of smartphone-based telematics for West Africa is not simply a matter of porting code from developed markets; it is an exercise in contextual engineering. The algorithms proposed in this report acknowledge the distinct physical and sociological realities of the region. By raising braking thresholds to account for aggressive traffic flow, utilizing pulse-width integration to distinguish pothole impacts from crashes, and employing gyroscopic variance to detect phone handling, we can create a robust safety profile.

The integration of these rule-based algorithms into a cohesive Android application offers a scalable, cost-effective solution to the region's road safety crisis. By democratizing access to driver feedback—placing a sophisticated event data recorder in the pocket of every *Trotro* and *Gbaka* driver—we move beyond punitive enforcement toward a culture of self-correction and data-driven safety.

Data Tables and Comparisons

Table 1: Smartphone Sensor Noise Characteristics (Sample Data) [5]

Smartphone Model	Sensor Axis	Bias Instability (m/s2)	Velocity Random Walk (m/s/Hz)

Samsung SM-A536V	Mean Accel	0.00133	0.00776
Google Pixel 7 Pro	Mean Accel	0.00021	0.00196
Vivo X60 Pro	Mean Accel	0.00034	0.00102
iPhone XR	Mean Accel	0.00088	0.00135
Implication: Higher noise in models like the Samsung SM-A536V requires more aggressive low-pass filtering (lower cutoff frequency) to prevent false positives in harsh braking detection.			

Table 2: Stopping Distance Estimations (Reaction + Braking) for Threshold Calibration [21]

Speed (km/h)	**Reaction Dist. (m)**	**Braking Dist. (m)**	**Total Stopping Dist. (m)**
40	17	9	26
60	25	20	45
80	33	36	69
100	42	56	98
Note: These values assume a standard friction coefficient. On West African roads with			

sand/gravel (reduced friction), braking distances increase, necessitating earlier and harder braking, justifying the higher g-force thresholds proposed in Section 7.2.			

Works cited

1. Africa Status Report on Road Safety - SSATP, accessed January 19, 2026, https://www.ssatp.org/sites/default/files/publication/SSATP_Africa%20Status%20Report%20on%20Road%20Safety%202025_single.pdf
2. Sensors Overview | Sensors and location - Android Developers, accessed January 19, 2026, https://developer.android.com/develop/sensors-and-location/sensors/sensors_overview
3. Motion sensors | Sensors and location - Android Developers, accessed January 19, 2026, https://developer.android.com/develop/sensors-and-location/sensors/sensors_motion
4. IMU-Kernel-110-120 Datasheet - Inertial Labs, accessed January 19, 2026, https://www.inertiallabs.com/wp-content/uploads/2023/07/IMU-Kernel-110-120_Datasheet.rev1_.8_Jul_2023.pdf
5. Smartphone MEMS Accelerometer and Gyroscope Measurement ..., accessed January 19, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC10490716/
6. An Automated Machine-Learning Approach for Road Pothole Detection Using Smartphone Sensor Data - PMC - PubMed Central, accessed January 19, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC7583950/
7. MIT Open Access Articles Smartphone Placement within Vehicles, accessed January 19, 2026, https://dspace.mit.edu/bitstream/handle/1721.1/136228/FINALVERSION.pdf?sequence=2&isAllowed=y
8. Sensing Vehicle Dynamics for Determining Driver Phone Use - WINLAB, Rutgers University, accessed January 19, 2026, https://www.winlab.rutgers.edu/~gruteser/papers/paper_CR.pdf
9. Effect of Circadian Rhythms and Driving Duration on Fatigue Level and Driving Performance of Professional Drivers - ResearchGate, accessed January 19, 2026, https://www.researchgate.net/publication/275855526_Effect_of_Circadian_Rhyth

ms_and_Driving_Duration_on_Fatigue_Level_and_Driving_Performance_of_Professional_Drivers
10. “Indiscipline” in context: a political-economic grounding for dangerous driving behaviors among Tro-Tro drivers in Ghana - ResearchGate, accessed January 19, 2026, https://www.researchgate.net/publication/342238374_Indiscipline_in_context_a_political-economic_grounding_for_dangerous_driving_behaviors_among_Tro-Tro_drivers_in_Ghana
11. Speed Bump Detection Using Accelerometric Features: A Genetic Algorithm Approach, accessed January 19, 2026, https://www.mdpi.com/1424-8220/18/2/443
12. Real Time Pothole Detection Using Android Smartphones with Accelerometers, accessed January 19, 2026, https://www.researchgate.net/publication/224253154_Real_Time_Pothole_Detection_Using_Android_Smartphones_with_Accelerometers
13. Bumps recorded using the mobile phone accelerometer. (a) Formation of a... - ResearchGate, accessed January 19, 2026, https://www.researchgate.net/figure/Bumps-recorded-using-the-mobile-phone-accelerometer-a-Formation-of-a-bump-with-an_fig7_260543538
14. DETECTING A VEHICLE CRASH USING A MOBILE COMPUTING DEVICE - Technical Disclosure Commons, accessed January 19, 2026, https://www.tdcommons.org/cgi/viewcontent.cgi?article=3642&context=dpubs_series
15. Car Accident Detection and Notification System Using Smartphone, accessed January 19, 2026, https://ijcsmc.com/docs/papers/April2015/V4I4201599a40.pdf
16. Accident and Roll Over Detection + Accident Data Exports - Digital Matter Knowledge Base, accessed January 19, 2026, https://support.digitalmatter.com/speeding-driver-behaviour/accident-detection-and-reconstruction
17. Accident Detection with Telematics - Dolphin Technologies, accessed January 19, 2026, https://www.dolph.in/podcast/accident-detection-with-telematics/
18. Driver's style classification using jerk analysis | Request PDF - ResearchGate, accessed January 19, 2026, https://www.researchgate.net/publication/224453294_Driver's_style_classification_using_jerk_analysis
19. Harsh Braking by Truck Drivers: A Comparison of Thresholds and Driving Contexts Using Naturalistic Driving Data - ResearchGate, accessed January 19, 2026, https://www.researchgate.net/publication/377665270_Harsh_Braking_by_Truck_Drivers_A_Comparison_of_Thresholds_and_Driving_Contexts_Using_Naturalistic_Driving_Data
20. A Geofence Violation Prevention Mechanism for Small UAS, accessed January 19, 2026, https://ntrs.nasa.gov/api/citations/20190000716/downloads/20190000716.pdf
21. Safe Distance | RKCL - Road Safety Web Portal, accessed January 19, 2026, https://roadsafetycell.rkcl.in/pageView/token=ckpqUWdrek1uYUdueDY3WnhHL3h

Fdz09
22. Driver Care: Know Your Stopping Distance - Automotive Fleet, accessed January 19, 2026, https://www.automotive-fleet.com/driver-care/239402/driver-care-know-your-stopping-distance