Peer-to-Peer Networking and Applications (2026) 19:120
https://doi.org/10.1007/s12083-026-02276-w

RESEARCH

Integrative framework for driver inattention detection and autonomous safety enhancement leveraging deep learning and blockchain network

Odinachi Udemezuo Nwankwo[1] · Muhammad Rasyid Redha Ansori[2] · Gifar Arif Haryadi[1,3] · Dong-Seong Kim[1,2,3] · Jae Min Lee[1]

Received: 15 January 2026 / Accepted: 30 June 2026

Abstract
This study introduces an integrated system for detecting and managing driver inattention using a combination of artificial intelligence (AI), blockchain technology, and edge computing. The proposed system utilizes You Look Only Once version 8 Nano (YOLOv8n) for real-time object detection and a Raspberry Pi 5 for edge-based analysis, ensuring efficient and accurate detection of inattentive behaviors such as drowsiness, distraction, and other unsafe driving patterns. Blockchain integration with Hyperledger Besu, using the Istanbul Byzantine Fault Tolerance version 2.0 (IBFT 2.0) consensus protocol, provides a secure and tamper-resistant ledger for recording and verifying driver inattention events. The system demonstrated robust performance metrics, including a mean precision of 92.99%, mean recall of 93.47%, and mean Average Precision at Intersection over Union threshold 0.5 (mAP@0.5) of 92.5% for YOLOv8n, establishing its superiority among evaluated YOLO models. The blockchain network achieved a throughput of up to 128.5 transactions per second (TPS) and an average latency ranging from 0.84 seconds for transfer operations to 4.43 seconds for open operations, demonstrating its capability to support real-time event logging. This research addresses limitations in traditional centralized systems by offering a scalable, permissioned, and transparent framework for enhancing driver safety through real-time monitoring and secure data management.

Keywords Deep learning · Driver inattention · Blockchain

1 Introduction

In recent years, the rapid advancements in Asia's industrial capabilities and technological sophistication have propelled significant social and economic progress. However, this growth has been accompanied by a substantial increase in vehicular traffic, particularly in countries like India and China, where transportation networks have expanded significantly. Consequently, road traffic accidents have become more frequent, posing substantial risks to daily commuting and overall safety. Among the various causes of these accidents, distracted driving emerges as a critical issue [29], with investigations revealing it as a primary factor, especially in severe incidents. Statistics indicate that distracted driving contributes to approximately 43% of accidents in heavy traffic and 37% on highways [27], particularly when drivers engage in activities such as drinking or using mobile phones, diverting their attention from the road. Such

✉ Jae Min Lee
ljmpaul@kumoh.ac.kr

Odinachi Udemezuo Nwankwo
odinachifoot@gmail.com

Muhammad Rasyid Redha Ansori
rasyidred@nslab.tech

Gifar Arif Haryadi
arpegio@kumoh.ac.kr

Dong-Seong Kim
dskim@kumoh.ac.kr

1 Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, Gyeongsangbuk-do, South Korea

2 NSLab Inc., Gumi 39177, Gyeongsangbuk-do, South Korea

3 ICT Convergence Research Center, Kumoh National Institute of Technology, Gumi 39177, Gyeongsangbuk-do, South Korea

Published online: 24 July 2026

behaviors, categorized as visual and cognitive distractions, significantly diminish drivers' decision-making and operational capacities, raising the likelihood of personal injury and property damage, particularly during high-speed driving [28]. Fatigue is also a significant concern, with drivers being considered fatigued after continuous driving for over four hours.

In crucial industrial operations, maintaining monitoring personnel's focus is vital to prevent accidents and subsequent material wastage and economic losses. Distraction and fatigue stand out as primary causes of lapses in attention. Typical distractions encompass looking-away, drinking, and mobile phone usage, with the latter posing significant risks. Both driving and industrial processes entail safety hazards, with fatigue and inattention impairing concentration [2].

Hence, to mitigate the likelihood of accidents, the installation of Advanced Driver Assistance Systems (ADAS) becomes crucial [15]. ADAS serves to alert drivers of potential risks or even intervene to avert collisions. Currently, both unmanned driving and ADAS technologies have gained widespread adoption. By identifying driver distractions, timely ADAS interventions prove instrumental in preventing numerous traffic accidents [19].

Currently, there are two primary approaches to assess personnel attention: wearable device-based detection and machine vision-based detection. Wearable device-based methods involve collecting physiological signals like electroencephalograms, [26] electrocardiograms, and electromyograms [8] through medical equipment, offering precise insights into the individual's physical state. However, these devices suffer from limited usability, complex operation, and high costs, potentially compromising the detection accuracy in practical settings. In contrast, machine vision-based methods analyze facial characteristics to assess attention divergence, offering a non-intrusive and cost-effective alternative [7]. This approach boasts superior real-time performance compared to wearable devices and is gaining traction, driven by advances in machine vision and deep learning technologies [13, 16]. In this work, the terms used to describe driver behavior are defined as follows for consistency. Drowsiness refers to fatigue-related loss of alertness characterized mainly by prolonged eye closure and yawning. Distraction refers to the diversion of visual or cognitive attention due to activities such as phone usage, drinking, texting, or looking away from the road. Inattention is used as a broader term that encompasses both drowsiness and distraction, representing any driver state that reduces focus on safe driving. These definitions are applied consistently throughout the manuscript.

Motivation Previous studies did not effectively address the issues associated with using a centralized Internet of Things (IoT) server for data storage. One major concern is that storing data on a central device, computer, or in the cloud makes it vulnerable to unauthorized alterations or loss. Earlier research did not adequately tackle the problem of data mutability, meaning that the stored data could be changed without proper authorization, potentially allowing malicious actors to tamper with the server's data.

In essence, the lack of comprehensive solutions in previous research left central storage systems like IoT servers and vehicle black boxes susceptible to data manipulation. The concentration of data in a single server made it an attractive target for data mutability, which could have far-reaching consequences for the system's integrity and functionality. The limited attention given to issues related to data mutability further exacerbated trust concerns. If data could be altered without proper oversight or safeguards, it significantly eroded the trustworthiness and accuracy of the stored information.

Prior research has not explored the integration of deep learning and blockchain technologies working in tandem for secure data storage. This novel research emphasizes the cohesive combination of these technologies, unlike previous studies that separately focused on either deep learning or blockchain. By harnessing their collaborative potential, this approach aims to enhance their capabilities and unlock new possibilities for applications that have not been explored before.

Contribution: The major contributions of this research are as follows:

1. This research enhances driver safety by developing an integrated detection and alert system using Raspberry Pi 5 for real-time monitoring of driver distractions and eye states. By incorporating Arduino Nano 33 IoT and a servo motor, the system provides immediate vibration and voice alerts, significantly improving upon traditional single-camera methods. Additionally, temporary delay of 3 seconds was introduced in the design to double-check the inattention class detected before confirmation in order to reduce false alarm rate.
2. An innovative framework was introduced for storing and managing critical driving behavior data, such as prolonged eye closures exceeding three seconds. A high-performance IBFT 2.0 Proof of Authority blockchain network was implemented to ensure secure and reliable data management without gas fees. Leveraging blockchain technology, the system enables safe driving incentives, post-accident investigations, and insurance premium adjustments by maintaining immutable driver performance records. Smart contracts were also developed to automate the recording of inattention events and manage secure access to this information.

3. To facilitate interaction with the system, a Flutter-based mobile decentralized application (DApp) was developed, featuring role-based access control. This interface allows various stakeholders, including police, road safety officials, insurance companies, and logistics organizations, to access driver performance data securely. The DApp ensures effective collaboration while safeguarding data privacy and integrity.

The subsequent sections of this document are structured as follows: Section 2 provides a comprehensive review of existing literature, highlighting gaps and motivations for the study. Section 3 presents the proposed system model, including its architecture, flowchart, and smart contract design. Section 4 outlines the dataset and performance metrics used to evaluate the system. Section 5 discusses the implementation and experimental results, including model performance and blockchain evaluation. Finally, Section 6 offers the conclusion of the paper and suggestions for future research directions.

2 Literature review

This section reviews three areas of related work that collectively motivate the proposed system. Figure 1 illustrates the broader ADAS context in which this work is situated. Although ADAS incorporates various sensors and communication modules for real-time environment perception, it cannot directly assess the cognitive state of the driver. Existing research has addressed this limitation through three primary directions: drowsiness detection, distraction detection, and blockchain-based data management. These domains define the technical context and highlight the gaps that the proposed system aims to address.

2.1 Drowsiness detection systems

A prominent thread of research focuses on detecting driver drowsiness through eye-state and facial cue analysis. One approach integrates near-infrared (NIR) imaging with deep learning. For example, Sinha et al. [22] combined NIR cameras with the YOLOv3 architecture for face detection, achieving 97% accuracy under partial occlusion. While effective in low-light conditions, the reliance on specialized hardware increases deployment cost, and the binary eye-state classification limits the ability to capture varying levels of drowsiness.

Lightweight and embedded solutions have also been explored. Soman et al. [24] proposed a real-time IoT framework using the Eye Aspect Ratio (EAR), deployed on a Raspberry Pi 4B with multimodal alert mechanisms. However, the use of a fixed EAR threshold (0.2) introduces sensitivity to individual facial differences and head pose variation, limiting generalizability. Similarly, Ziryawulawo et al. [30] demonstrated a camera-based system in a real-world transit setting, confirming deployment feasibility, but did not validate performance across diverse environments or driver populations.

Deep learning-based approaches provide higher accuracy but introduce computational challenges. Phan et al. [21], a hybrid system combining facial landmark detection with MobileNetV2 and ResNet-50 achieved 97% accuracy. However, the high parameter count of ResNet-50 raises concerns for edge deployment, and real-time performance metrics were not reported. Ganguly et al. [11] proposed two-stage pipeline using Faster R-CNN followed by a CNN classifier, achieved 97.6% accuracy but increases inference latency due to sequential processing.

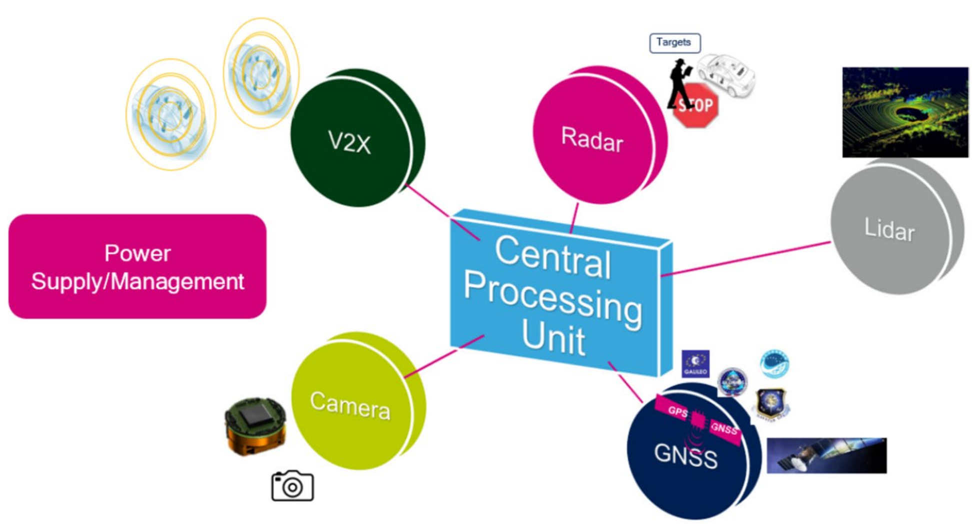

Fig. 1 Overview of Advanced Driver Assistance Systems (ADAS) technology illustrating the integration of multiple sensing and communication modules—including Radar, Lidar, Camera, (Global Navigation Satellite Systems (GNSS), and Vehicle-to-Everything (V2X)—coordinated through a Central Processing Unit (CPU). The system relies on Power Supply/Management for sustained operation and supports real-time environment perception and decision-making to enhance vehicle safety and automation

Efficient object detection frameworks have also been considered. Shakeel et al. [23] employed MobileNet with SSD, achieving an mAP of 0.84 on embedded devices such as Raspberry Pi 3 and smartphones. While computationally efficient, the accuracy may be insufficient for safety-critical applications, and binary outputs cannot capture gradual drowsiness onset.

2.2 Driver distraction detection systems

A parallel body of work addresses driver distraction through activity classification. Christy et al. [6] used CNN models to classify seven distraction scenarios under stress conditions, achieving 99% accuracy. However, the dataset included only ten drivers, limiting generalizability and lacking cross-subject validation.

Comparative analyses of deep learning architectures have also been conducted. Hossain et al. [12], MobileNetV2 achieved the highest accuracy (98.12%) among evaluated models. Despite this, the study focused solely on accuracy without considering deployment constraints such as inference latency or memory usage, which are critical for real-time systems.

Benchmark-driven approaches have shown strong performance under controlled conditions. Bekka et al. [3] utilized a VGG16-based model trained on the State Farm Distraction Dataset, achieving over 99% accuracy with real-time alert functionality. However, VGG16's large computational footprint limits its applicability in resource-constrained environments, and the dataset's staged nature reduces its representativeness of real-world driving behavior.

2.3 Blockchain-based data management frameworks

A third line of research explores blockchain for secure and decentralized data management, primarily in the healthcare domain. Zhang et al. [31] enables secure clinical data exchange between patients and providers, while Patel [20] proposed off-chain mechanisms to improve scalability in medical data sharing. Similarly, Usman and Qamar [25] introduced a private blockchain for electronic medical records, and Du et al. [9] suggested a consortium-based model where hospitals maintain shared data infrastructure.

Smart contract-based solutions have also been explored to enhance access control and process efficiency. For instance, Khatoon [14] demonstrated improved healthcare workflows through automated data access management, although overly permissive access policies may introduce security risks. More recently, Aditya et al. [1] proposed a comparable integration of deep learning classification with blockchain smart contracts and cryptographic security for IoT-based healthcare monitoring, demonstrating strong performance in both threat detection and disease diagnosis.

Despite these advances, blockchain applications remain largely confined to healthcare. None of the reviewed frameworks have been adapted to vehicular monitoring systems, leaving a gap between decentralized data management capabilities and their application in driver safety contexts.

2.4 Research gaps and motivation

Across the reviewed studies, two structural gaps consistently emerge. First, drowsiness detection and distraction detection have evolved independently. Eye-state monitoring systems do not account for cognitive or manual distractions, while activity-based systems do not incorporate physiological indicators of fatigue. No existing work integrates both modalities into a unified detection framework.

Second, current driver monitoring systems rely on centralized data architectures, which introduce single points of failure, limit auditability, and restrict secure multi-stakeholder access. While blockchain-based systems demonstrate the feasibility of decentralized data management, this approach has not been extended to vehicular environments.

These gaps collectively motivate the proposed system, which aims to unify driver inattention detection and incorporate decentralized data management within a single framework. Table 1 summarizes the reviewed studies against key evaluation criteria.

3 Problem formulation

We aim to detect driver inattention and securely store the detected information in a decentralized database. The problem consists of two components: **Inattention Detection** and **Decentralized Storage**.

Table 1 Comparison of studies on driver inattention detection

Studies	Distraction	Drowsiness	Decentralized Access	Real-time Alert
Sinha et al. [22]	×	✓	×	✓
Soman et al. [24]	×	✓	×	✓
Ziryawulawo et al. [30]	×	✓	×	✓
Phan et al. [21]	×	✓	×	✓
Ganguly et al. [11]	×	✓	×	✓
Shakeel et al. [23]	×	✓	×	✓
Christy et al. [6]	✓	×	×	×
Hossain et al. [12]	✓	×	×	✓
Bekka et al. [3]	✓	×	×	✓
Proposed Tech.	✓	✓	✓	✓

3.1 1. Inattention Detection

Driver inattention (I) is influenced by **drowsiness** (D) and **distraction** (R):

Input Variables:

- $\mathrm{X} \in \mathbb{R}^n$: Feature vector representing observed driver behavior, where n is the number of observed features.**Weight Matrices:**

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \dots & d_{mn} \end{bmatrix}, \quad R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{bmatrix}.$$

Inattention Score:

$$I = \alpha D\mathrm{X} + \beta R\mathrm{X},$$

where α and β are weights to balance the influence of drowsiness and distraction.

3.2 2. Decentralized Storage

Detected inattention (I) is stored in a decentralized database (Z) for future use.

Database Representation:

$$\mathrm{Z} = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{m1} & z_{m2} & \dots & z_{mk} \end{bmatrix},$$

where each row Z_i contains:

$$\mathrm{Z}_i = (t_i, I_i, c_i, l_i),$$

with:

- t_i: Timestamp.
- I_i: Inattention score.
- c_i: Confidence level.
- l_i: Location or metadata.

3.3 3. Objective

Minimize the overall error (E) in detecting and storing inattention:

$$E = \|I - Y\|^2,$$

where Y is the ground truth for inattention labels.

3.4 4. Constraints

Real-time Processing:

$$T(I) \leq \tau,$$

where $T(I)$ is the time to compute I, and τ is the maximum allowable time.

Secure Storage:

Z_i must be stored immutably in the blockchain.

It is important to clarify that the above mathematical formulation serves as a conceptual abstraction describing the relationship between drowsiness, distraction, and overall driver inattention. The proposed YOLO-based implementation does not explicitly compute the weight matrices or the inattention score as expressed symbolically. Instead, these relationships are implicitly learned by the deep neural network during training through convolutional feature extraction and classification. The formulation is therefore intended to provide theoretical intuition for the problem structure rather than to represent the exact computational pipeline of the implemented system.

4 Proposed system

The proposed system model is designed to detect and manage driver inattention in real time using a combination of artificial intelligence, edge computing, blockchain, and decentralized applications (DApps). The model is divided into two main phases: Design and Implementation, which work together to provide an efficient, real-time, and transparent monitoring system.

4.1 System architecture

Figure 2 illustrates the overall architecture of the proposed driver inattention detection framework, which is organized into two complementary phases: the design phase and the implementation phase. The design phase presents the conceptual workflow, where a vehicle-mounted camera captures driver behavior, an edge node performs AI-based object detection, a blockchain server securely stores confirmed inattention events, and a decentralized application (DApp) enables authorized stakeholders to access the recorded information. The implementation phase demonstrates the practical realization of this architecture using a Raspberry Pi 5 running YOLOv8n for edge inference, a Hyperledger Besu blockchain network for immutable event logging, and a Flutter-based DApp for secure visualization and

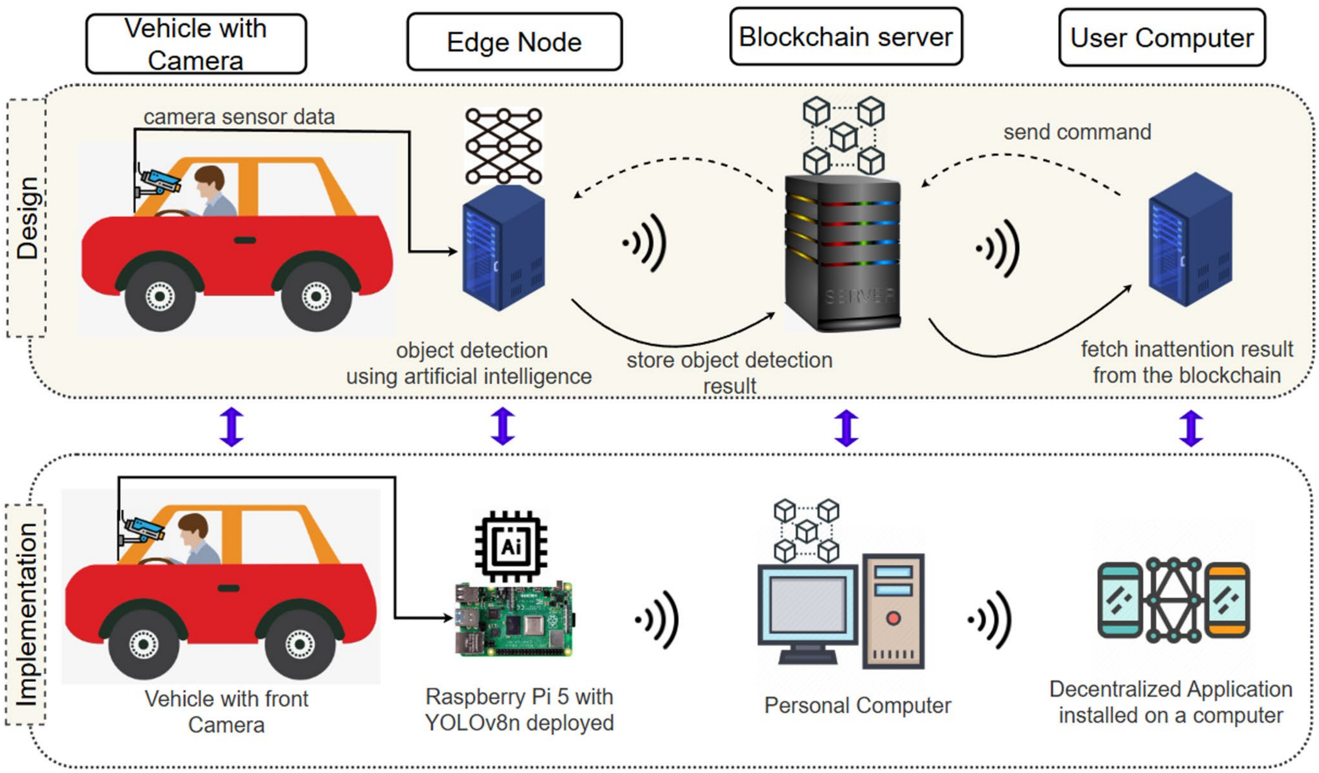

Fig. 2 Proposed system model showcasing both the design and implementation phases of a blockchain-integrated object detection system. The model includes a vehicle equipped with a camera capturing real-time footage, an edge node (Raspberry Pi 5) running YOLOv8 for AI-based object detection, a blockchain server for securely storing detection results, and a user computer with a decentralized application (DApp) to access and manage inattention alerts

management of driver inattention records. This layered architecture bridges the conceptual system design with its real-world deployment, highlighting the integration of artificial intelligence, edge computing, and blockchain technologies into a unified driver monitoring framework.

The framework begins with a vehicle-mounted camera that captures real-time driver behavior. The captured data is transmitted to an edge node for local processing. The YOLO model in the proposed system performs object detection only, producing bounding boxes and class labels for driver behaviors. The interpretation of these detections into driver state classification (normal, distracted, or drowsy) is handled by a separate rule-based temporal verification module. Detected behaviors such as eye closure, yawning, phone usage, drinking, and looking away are tracked across consecutive frames, and a three-second persistence check is applied before confirming true inattention. This approach does not compute a numerical inattention score; instead, behavioral state is determined based on event persistence over time. This approach is more suitable for real-time edge deployment in vehicular environments. The edge node identifies inattentive states, such as distraction and drowsiness, and transmits the inference results to a Hyperledger Besu blockchain for immutable storage. To ensure privacy-preserving accountability, driver identities are encoded via 160-bit Ethereum-compatible wallet addresses, acting as pseudo-anonymous identifiers that shield personal credentials. To maintain ledger scalability, the system employs a hybrid storage model where high-bandwidth raw data is discarded post-inference at the edge, while only lightweight metadata, comprising the event class and timestamp, is committed on-chain.

The blockchain server also facilitates interactions with a user's computer. The user computer retrieves inattention detection results from the blockchain. The user computer can analyze these results or issue commands based on the data. For example, it might generate alerts or provide recommendations to improve driver safety. This blockchain-based architecture enables seamless communication between all components of the system and supports real-time monitoring and decision making.

The implementation phase transforms the conceptual design into a practical system. A vehicle with a front camera collects real-time footage of the driver, focusing on capturing signs of inattention. The captured data is processed locally using a Raspberry Pi 5, which runs the YOLOv8n object detection algorithm. This lightweight and high-performance edge device ensures real-time detection without heavy reliance on external computational resources, thereby reducing both latency and dependency on network connectivity.

Once the analysis is complete, the detection results are sent to a personal computer that acts as a central hub for further processing and visualization. This computer integrates with a decentralized application (DApp), which provides a user-friendly interface for interacting with the blockchain. The DApp allows users to verify the authenticity of results, track driver behavior records, and monitor driving safety insights. By utilizing a DApp, the system ensures secure and tamper-proof interaction, enhancing data accessibility and user confidence in the results. The DApp retrieves these records by querying indexed Solidity events stored on the Hyperledger Besu network. This indexing structure allows stakeholders to filter millions of events by driver address or inattention type through efficient indexed event lookups, which provides the scalability required for large-scale fleet monitoring.

Figure 3 depicts the proposed flowchart for a real-time system for detecting and addressing drivers' inattentive behaviors. It starts with the vehicle's camera capturing images. These images are then analyzed by a deep learning model on the edge node. If unsafe behavior is detected, the system triggers an initial alarm and monitors for distraction or prolonged eye closure, with re-evaluations and escalated alerts as needed. Confirmed inattentive behaviors are securely stored on a blockchain, enabling data access through DApps for stakeholders such as companies, insurance providers, and traffic authorities, ensuring transparency and actionable insights. Three seconds delay was chosen because eye closure lasting for more than two seconds is considered to be drowsiness [4, 5, 17].

The pseudocode in algorithm 1 describes a system that detects and manages inattentive driver behaviors using YOLO object detection and logs the results on the Ethereum blockchain. It begins by initializing the test to speech (TTS) engine, connecting to the Ethereum network via Infura, and loading YOLO model weights and configurations. The system enters a loop where it captures video frames from a camera, performs YOLO detection to identify unsafe behaviors (e.g., drinking, texting, yawning), and triggers alarms if such behaviors are detected. If behaviors persist after a recheck,

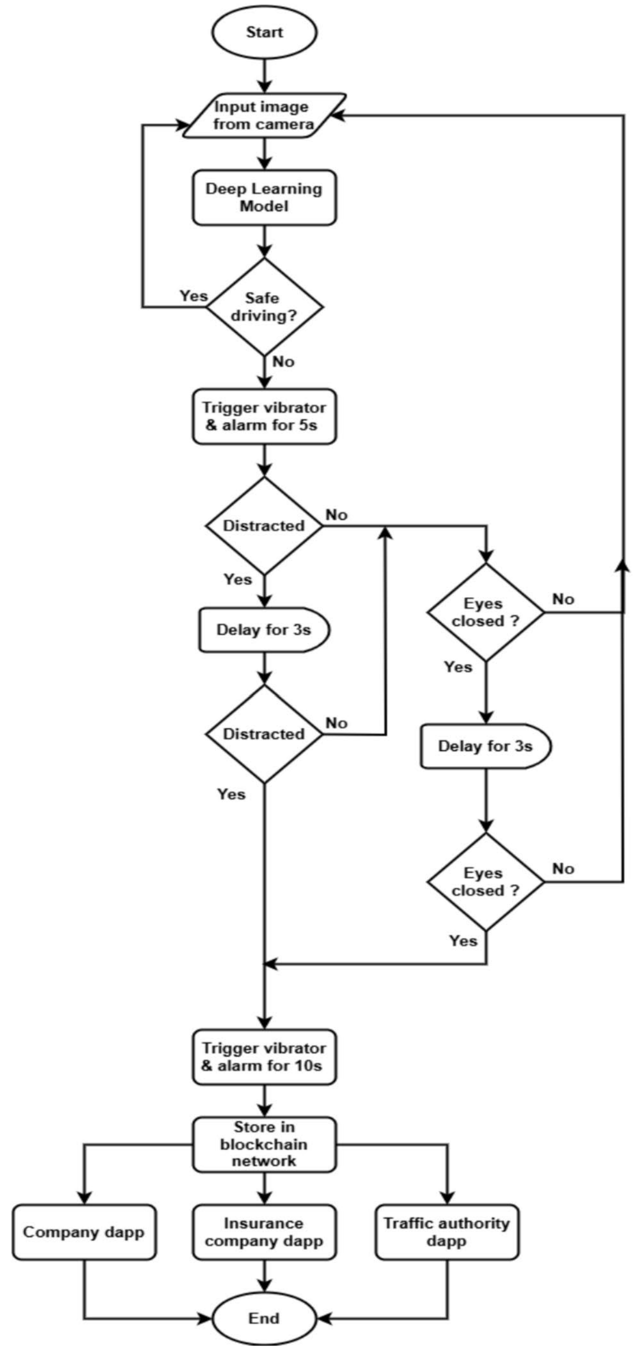

Fig. 3 The Flowchart of the proposed system

the system categorizes them as either "drowsy" or "distracted" and logs the incidents on the blockchain by creating and signing a transaction with the driver's private key. The process continues until manually exited, ensuring real-time detection, escalation, and secure data logging for transparency and accountability.

Algorithm 1 Inattention Driver Detection System with Blockchain Logging

Input: $addr_{driver}$: driver Ethereum address, PK: private key
Output: Blockchain record of inattention events
Notations:
  B: Bounding boxes, L: Labels, C: Confidence,
  D: Drinking, L_A: Looking away, T_P: Talking on phone,
  C_E: Closed eyes, T: Texting, Y: Yawning, TX: Transaction
Initialize TTS engine and connect to the IBFT 2.0 blockchain network via RPC
if connection successful **then**
  Announce: "Connected"
else
  Announce: "Connection failed" and exit
end if
Load YOLO weights, config, and labels
Start camera feed and initialize serial communication
while true **do**
  Capture and resize frame
  $\{B, L, C\} \leftarrow$ Perform YOLO detection
  if $L \cap \{D, L_A, T_P, C_E, T, Y\} \neq \emptyset$ **then**
    Trigger alarm and announce: "Unsafe driving"
  end if
  if L contains $\{D, L_A, T_P, T, Y\}$ **then**
    Wait 3 seconds and recheck detection
  end if
  if L still contains $\{D, L_A, T_P, T, Y\}$ **then**
    Trigger alarm and announce: "Distracted". Call submit(distraction)
  end if
  if L contains C_E **then**
    Wait 3 seconds and recheck detection
  end if
  if L still contains C_E **then**
    Trigger alarm and announce: "Sleeping". Call submit(drowsy)
  end if
  Display processed frame with B
  if Esc key pressed **then**
    break
  end if
end while
procedure SUBMIT(inattentionType)
  if inattentionType = drowsy **then**
    $TX \leftarrow$ recordDrowsyDriver($addr_{driver}$)
  else
    $TX \leftarrow$ recordDistractionDriver($addr_{driver}$)
  end if
  $SignedTX \leftarrow$ Sign(TX, PK)
  Send transaction and wait for receipt
  Announce "Logged" and return transaction hash
end procedure

4.2 Blockchain network

The system's blockchain infrastructure was implemented using Hyperledger Besu, a permissioned blockchain infrastructure designed for enterprise-grade applications. The blockchain network was configured with the Istanbul Byzantine Fault Tolerance Protocol (IBFT 2.0), a PoA variant specifically designed for authorized blockchain networks [10]. IBFT 2.0 ensures consensus through a predefined set of authorities, providing improved resilience against Byzantine faults while maintaining network coherence and reliability. The IBFT 2.0 consensus mechanism is particularly suitable for the system's requirements, as it allows trusted validators to maintain network integrity in a permissioned environment. The network creation and consensus process is illustrated in Fig. 4, and the interface of the node's operational status is presented in Fig. 5.

The system's blockchain infrastructure was implemented using Hyperledger Besu (v25.11.0), a permissioned blockchain client designed for enterprise-grade applications. The network is configured around four validator nodes ($n = 4$), with one node assigned per institutional stakeholder, namely Police, Road Safety Authority, Insurance Companies, and Logistics Organizations. This validator count simultaneously satisfies the IBFT 2.0 minimum Byzantine fault-tolerant network size ($n \geq 3f + 1$ for $f = 1$) and aligns with the four-party governance structure inherent to the system design. Additional validator nodes beyond $n = 4$ would introduce $O(n^2)$ consensus-message overhead without a corresponding governance justification, as each consortium party is already represented. Besu operates two parallel networking stacks. The first is a UDP-based discovery stack for finding peers, and the second is a devP2P stack over TCP for all data transport once peers are bonded. In this deployment, the UDP discovery stack is disabled (`-discovery-enabled=false`) and connectivity relies on a static node list (`static-nodes.json`), in which each validator holds the enode URLs of the other three; this prevents the node from soliciting connections from unknown peers. Note that `static-nodes.json` governs outbound connectivity to the known validator set, and the experimental deployment operates on a private local network not exposed to public internet access. This design prioritises security and determinism over dynamic scalability: restricting connections to a known validator set reduces the attack surface against Sybil and eclipse attacks, at the cost of limiting open participation. The result is a logically fully connected mesh of persistent TCP devP2P connections among the four validators. Edge devices (Raspberry Pi) and the Flutter DApp do not act as validators; they participate as lightweight JSON-RPC clients that submit locally signed transactions to their respective institutional validator's RPC endpoint over HTTP/WebSocket.

Network configuration Table 2 summarises the `genesis.json` parameters governing the experimental network. The block period of 1 second provides rapid deterministic finality;

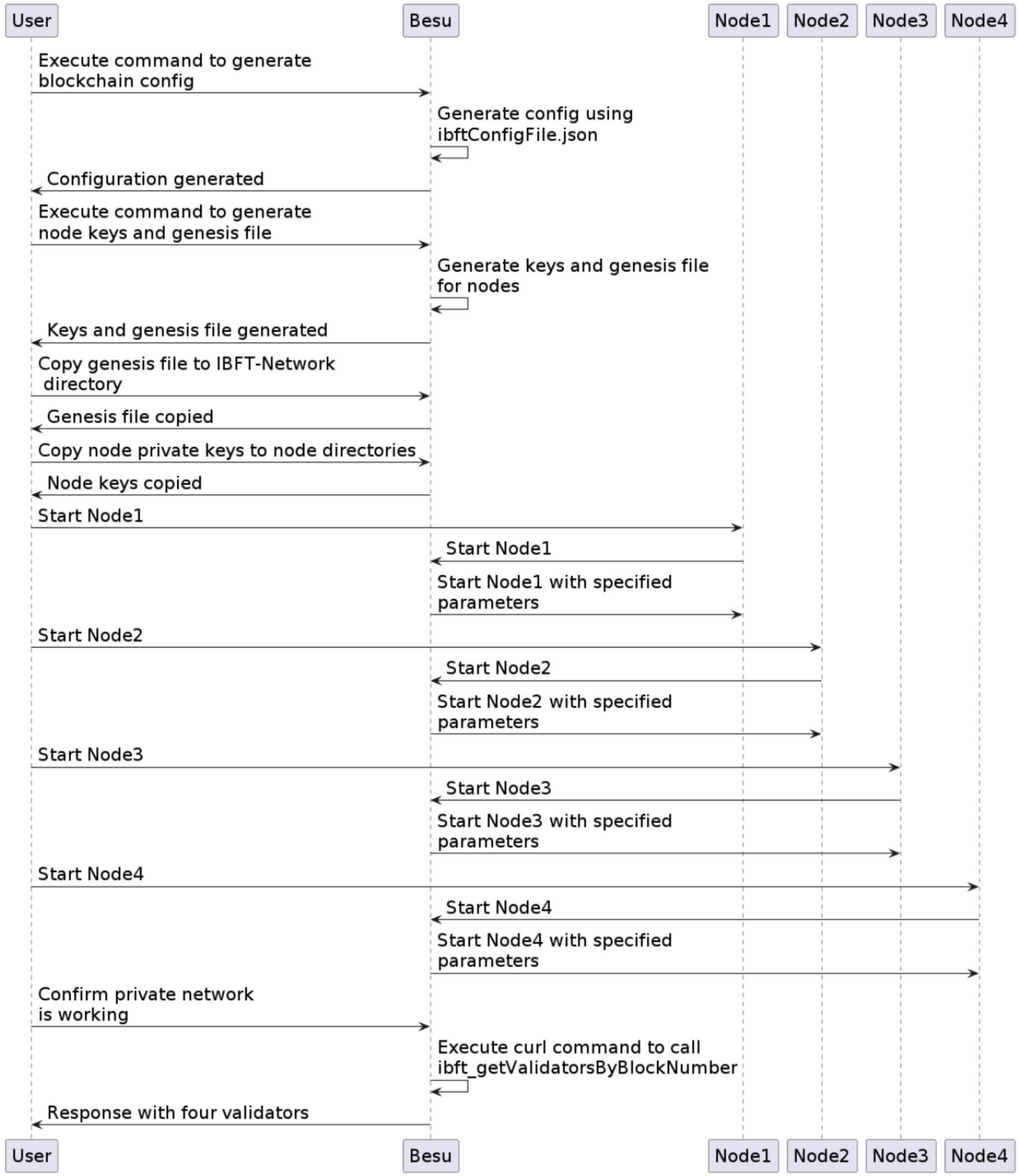

Fig. 4 Network Setup for IBFT 2.0

the request timeout of 4 seconds ensures a round-change is triggered only when the current proposer is genuinely unresponsive rather than delayed by normal propagation. The gas price is fixed at zero, enabling a fee-free institutional environment. The gas limit of 4,700,000 was dimensioned relative to the cost of the core write operations. The `recordDrowsyDriver()` and `recordDistractionDriver()` functions each perform one

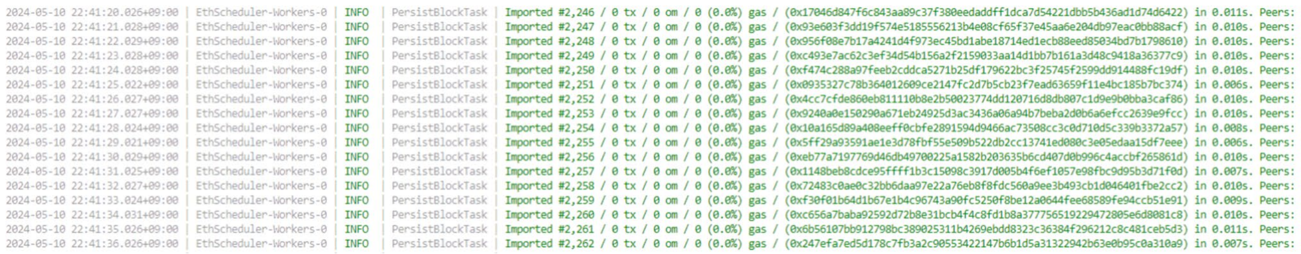

Fig. 5 Zero-Gas-Fee Blockchain Network Activated

Table 2 Hyperledger Besu IBFT 2.0 Network Configuration

Parameter	Value	Configuration field
Besu version	25.11.0	—
Consensus protocol	IBFT 2.0	`consensus`
Validator count	4	`ibft2.validators`
Block period	1 s	`blockperiodseconds`
Request timeout	4 s	`requesttimeoutseconds`
Gas price	0	`-min-gas-price=0`
Gas limit	4,700,000 (0x47b760)	`gasLimit`
Fault tolerance f	1	$\lfloor (4-1)/3 \rfloor$
Node topology	Fully connected mesh	`static-nodes.json`

to two `SSTORE` operations and emit a Solidity event, consuming approximately 30,000–50,000 gas per transaction. This yields a per-block write capacity of 94–157 operations, which directly accounts for the observed write-throughput ceiling of 94–128 TPS; the gas limit is correctly matched to the IBFT 2.0 throughput envelope at a 1-second block interval and is not the performance bottleneck, as the limiting factor is consensus round-time. IBFT 2.0 was selected for its deterministic finality and well-established performance characteristics at small validator sets.

Threat model The primary adversary is an insider with administrative access to a centralised logging system, such as a database administrator at a logistics company attempting to delete a driver's drowsiness record to avoid liability after an accident. A conventional relational database cannot fully mitigate this threat, as a sufficiently privileged administrator can disable audit triggers, restore from a tampered backup, or rotate credentials. The audit-log protection chain terminates at a trusted human. The proposed blockchain architecture addresses this by distributing write authority across four institutionally independent validators. Appending any block to the chain requires the commit quorum of $2f + 1 = 3$ validators to sign it. A malicious actor controlling one node cannot unilaterally alter or delete any confirmed record without colluding with at least two other independent stakeholders that have competing legal and financial interests. The security guarantee combines cryptographic consensus with institutional separation of authority, which is the standard trust model for permissioned consortium blockchains.

Fault tolerance and recovery IBFT 2.0 guarantees safety, meaning no conflicting blocks are ever finalised, provided fewer than $f + 1 = 2$ validators are simultaneously Byzantine. If the current block proposer fails to broadcast a PRE-PREPARE message within the 4-second request timeout, the remaining validators automatically initiate a round-change and elect a new proposer, resuming block production without manual intervention. As documented in Section 8, liveness requires at least $2f + 1 = 3$ validators to be concurrently online; the implications of this boundary are discussed there.

The latency between event submission and confirmed on-chain record is decomposed as:

$$T_{\text{blockchain}} = T_{\text{network}} + T_{\text{consensus}} \tag{1}$$

where T_{network} is the JSON-RPC submission delay and $T_{\text{consensus}}$ is the IBFT 2.0 block finalisation time. The minimum consensus latency is bounded by:

$$T_{\text{consensus}} \geq T_{\text{block\_period}} + 2 \cdot \text{RTT}_{\text{validator}} \tag{2}$$

With a 1-second block period and co-located validators ($\text{RTT}_{\text{validator}} \approx 0$), $T_{\text{consensus}}$ approaches 1 second as a lower bound. Blockchain submissions are handled asynchronously after the inattention event is confirmed locally, so that $T_{\text{blockchain}}$ does not affect driver safety response time.

4.3 Smart contract

The smart contract, which serves as the backbone of the system's blockchain integration, was developed using the Solidity programming language, a widely adopted choice for creating self-executing contracts on blockchain platforms. The contract was subsequently deployed to the Hyperledger Besu network. The implemented smart contract comprises four essential functions, each serving a specific purpose in the overall system architecture. These functions are as follows.

- recordDrowsyDriver(): This function is invoked to record instances of detected drowsiness driving. When the deep learning-based detector identifies a driver that exhibits signs of drowsiness, this function is activated to store the relevant information on the blockchain.
- recordDistractionDriver(): When the detector spots a driver engaging in distracted behavior, whether they are texting, calling, drinking, or not focusing on the road, this function saves such actions to blockchain.
- getDrowsyDriverEvents(): This function allows authorized parties to retrieve all recorded drowsy driving events from the blockchain.
- getDistractionDriverEvents(): This function facilitates the retrieval of blockchain-recorded distractions from authenticated organizations.

The smart contract was further hardened through four deliberate design decisions. **Key management** is handled entirely at the edge: the Raspberry Pi autonomously signs transactions using an offline-generated private key stored in isolated environment variables, ensuring credentials never traverse the network. **Authorization granularity** is enforced through a registry contract implementing Role-Based Access Control (RBAC), so only verified stakeholders (e.g., police, insurance, road safety, and logistics entities) can execute state-changing functions or retrieve sensitive records. **Event auditing** is achieved via immutable Solidity `event` emissions that log behavior classes and timestamps on-chain post-transaction, enabling the DApp to reconstruct driver histories through gas-free, efficient indexed off-chain queries without redundant state enumeration. Finally, **vulnerability mitigation** was performed through static analysis via Slither and Aderyn, which collectively checked over 130 known attack vectors and confirmed the absence of reentrancy and arithmetic overflow vulnerabilities.

4.4 Practical deployment considerations

For real-world applicability, it is important to address practical deployment challenges such as data privacy, false alarm mitigation, and compatibility with existing Advanced Driver Assistance Systems (ADAS). The proposed framework was designed with these considerations in mind to ensure safe and effective operation in realistic vehicular environments.

4.4.1 Data privacy preservation

The system adopts a privacy-preserving architecture by ensuring that all video processing and driver behavior analysis occur locally on the edge device (Raspberry Pi 5). Raw images and video streams are neither transmitted over the network nor stored externally. After inference, only lightweight metadata consisting of the detected inattention event type and timestamp is committed to the blockchain. Drivers are represented using pseudo-anonymous wallet addresses rather than personal identifiers. The permissioned blockchain network implemented with Hyperledger Besu further restricts access through role-based permissions, allowing only authorized institutions such as police, insurance organizations, road safety authorities, and logistics companies to retrieve relevant records. The edge-processing and blockchain-based storage approach ensures accountability while minimizing exposure of sensitive personal data.

4.4.2 False alarm mitigation

To reduce false alarms caused by transient driver movements such as brief glinks, blinking, or posture adjustments, the system incorporates a temporal verification mechanism. When an unsafe behavior is first detected, the system waits for three seconds and performs a second evaluation before confirming the event. Only persistent behaviors are classified as genuine distraction or drowsiness and subsequently logged on the blockchain. In addition, the use of multi-class object detection (e.g., phone usage, drinking, looking away, eye closure, yawning) provides contextual validation, reducing reliance on a single visual cue and improving detection robustness.

4.4.3 Integration with existing ADAS platforms

The proposed system is designed as a modular driver monitoring and accountability layer that can operate alongside existing ADAS infrastructures without altering their core sensing components such as radar, LiDAR, GNSS, or V2X modules. Alerts generated by the system can be delivered through standard audio or visual channels already present in vehicles. Since inference is performed locally on the edge device and blockchain logging occurs asynchronously, the framework does not introduce additional latency into the real-time vehicle control loop. This design enables straightforward integration as an auxiliary safety and accountability module within current ADAS architectures.

5 Dataset description

This section presents the dataset that is used for detecting driver inattention, followed by the performance metrics used to evaluate the proposed system.

5.1 Dataset collection

The proposed system adopted an object detection-based computer vision approach to solving the driver inattention problems. Instances of driver inattention identified as nine classes namely Drinking, Driver Looking away, Driver talking on Phone, Driver, Eyes Closed, Eyes Open, Seat Belt, Texting and Yawning made-up input RGB image dataset. The 640 by 640 pixel input dataset from Roboflow [18] which comprises of 3787 images for training, 1078 images for validation and 550 images for testing were utilized to train, validate and test Deep learning-based computer vision models such as YOLOv5n (Nano), YOLOv6n (Nano), YOLO V7 and YOLOv8n (Nano) for object detection task. It is important to note that the dataset does not provide explicit metadata regarding the number of unique drivers, demographic distribution, or exhaustive real-world variations such as extreme lighting conditions, complex occlusions, and diverse camera placements. The dataset was primarily used to validate the feasibility of the proposed detection framework rather than to claim comprehensive real-world generalization.

5.2 YOLO training configuration and reproducibility

To ensure experimental transparency and full reproducibility of the object detection results, the training configuration, hyperparameters, and data augmentation procedures adopted for all YOLO models are explicitly documented in this section. All experiments were conducted using the Ultralytics implementation of YOLOv8, while maintaining identical training conditions for YOLOv5n, YOLOv6n, YOLOv7, and YOLOv8n to guarantee a fair performance comparison.

5.2.1 Training hyperparameters

The models were trained using the stochastic gradient descent (SGD) optimizer with momentum. The training parameters were selected based on convergence stability and object detection practices for 416×416 RGB inputs. Table 3 summarizes the adopted hyperparameters.

The choice of 50 epochs was guided by empirical convergence analysis. The validation mean Average Precision (mAP) for all YOLO variants stabilized after approximately 50 epochs, and further training did not yield significant improvements, indicating convergence without overfitting.

Table 3 YOLO Training Hyperparameters

Parameter	Value
Input image size	416×416
Batch size	8
Number of epochs	50
Initial learning rate	0.01
IoU threshold (evaluation)	0.5
Confidence threshold	0.25

5.2.2 Data augmentation strategy

The default data augmentation pipeline provided by Ultralytics YOLO was utilized during training. Rather than manually configuring individual augmentation operations, the training process relied on the framework's built-in augmentation strategy and default parameter settings, which are internally optimized for object detection tasks.

5.2.3 Reproducibility procedure

The experiments can be reproduced by following these steps:

1. Obtain the dataset described in Section 4.1 from Roboflow with the same class labels and image resolution.
2. Install the Ultralytics YOLO framework.
3. Train the models using the documented hyperparameters and input size.
4. Evaluate using an IoU threshold of 0.5 and a confidence threshold of 0.25.

A representative training command is provided below for clarity:

```
yolo detect train data=driver.yaml model
=yolov8n.pt imgsz=416 epochs=50 batch=8
```

By maintaining identical training settings across all YOLO variants, the comparative evaluation presented in this study remains fair, transparent, and fully reproducible.

6 Performance metrics

To comprehensively evaluate the effectiveness and efficiency of the proposed driver inattention detection system, two categories of performance metrics were employed: object detection metrics to assess the precision of the deep learning models, and blockchain network metrics to measure system responsiveness.

6.1 Object detection metrics

Several key metrics were used to evaluate various aspects of the performance of the detection model, from its ability to accurately locate objects to its precision in classification tasks.

6.1.1 Mean average precision (mAP)

Mean Average Precision serves as the primary metric for evaluating object detection performance, measuring the model's ability to make accurate detections with significant overlap between predicted and actual bounding boxes. For this study, mAP at an IoU threshold of 0.50 (mAP@50%) can be determined using:

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i \tag{3}$$

The Average Precision for each class (AP_i) is obtained by integrating over the precision-recall curve following:

$$AP = \int_0^1 P(R)\, dR \tag{4}$$

6.1.2 Precision (P)

To measure the model's ability to avoid false positives, precision is calculated using:

$$P = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \tag{6}$$

6.1.3 Recall (R)

The effectiveness in identifying actual positive cases is quantified through:

$$R = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{7}$$

6.1.4 F1-Score

To balance precision and recall in a single metric, the F1-Score is derived from:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{8}$$

6.1.5 Intersection over union (IoU)

The accuracy of bounding box predictions is measured through the overlap ratio given by:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{9}$$

6.2 Blockchain network metrics

To assess the practical viability of blockchain implementation, two critical performance aspects were measured: the system response time and its transaction processing capabilities.

6.2.1 Latency

Network responsiveness is assessed by measuring the time interval between submission and confirmation, which follows:

$$Latency = TCT - TST \tag{10}$$

where *TCT* represents the transaction confirmation time and *TST* denotes the transaction submission time. Note that this operational measurement formula complements the architectural decomposition given in Eq. 1, which models the theoretical lower bound of end-to-end latency.

6.2.2 Throughput

The network's transaction processing capacity is measured as the number of confirmed transactions per unit time.

$$Throughput = \frac{CT}{t} \tag{11}$$

where *CT* represents the number of confirmed transactions and *t* is the measurement duration.

7 Implementation and experimental results

7.1 Implementation

Figure 6 shows the prototype of our proposed solution using Raspberry Pi 5, a Rapoo USB Camera, speakers, Arduino Nano 33 IoT, and a servo motor. The camera monitors the driver's face in real-time while the Raspberry Pi detects inattention. Upon detection, it communicates with the Arduino to trigger the servo motor to vibrate the driver's seat belt for 10 seconds and outputs a warning voice alarm through the speakers to alert the inattentive driver.

Figure 7 presents the Android-based decentralized application (DApp) developed using Flutter, which serves as an interface for various road safety stakeholders to access and monitor driver inattention records. The DApp implements a role-based access control system, allowing four distinct stakeholder categories—police authorities, road safety officials, insurance companies, and logistics companies—to query driver records using blockchain addresses. The blockchain integration, demonstrated through the smart contract interface in Fig. 8, provides the underlying infrastructure for the DApp's functionality.

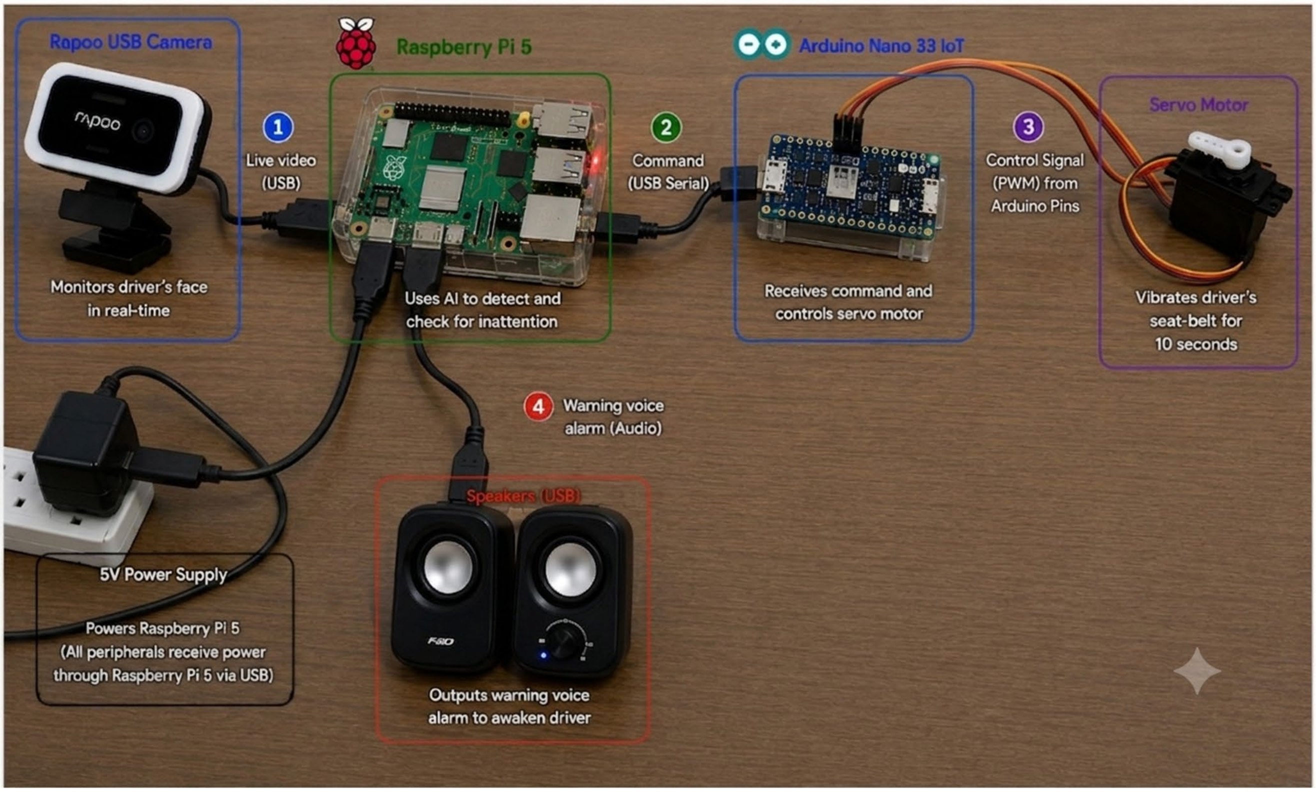

Fig. 6 A working prototype of the design using Raspberry pi 5, Rapoo USB Camera, a pair of speakers, Arduino nano 33 IoT, and a Servo motor. The camera monitors the drivers' face in real-time while the Raspberry Pi 5 uses its intelligence and input from camera to detect and checks for inattention, upon detection, it communicates to the Arduino to trigger the servo-motor for 10 seconds to vibrate driver's seat-belt, the speakers outputs the warning voice alarm to awaken inattentive driver

To manage the time required for blockchain transactions to be confirmed, the Flutter DApp is designed asynchronously so that the application never freezes while waiting for data. When stakeholders check a driver's history, the DApp fetches the records in the background by listening to Solidity `events`. Such background fetching keeps the user interface smooth and responsive. Most importantly, the real-time safety alarms inside the vehicle operate entirely on the edge device. Such local execution ensures that the asynchronous blockchain logging never delays the split-second warnings required for driver safety.

7.2 YOLO model performance

The performance of different YOLO object detection models (YOLOv8n, YOLOv7, YOLOv6n, and YOLOv5n) was evaluated based on several key metrics: mean precision, mean recall, mean F1-score, and mean Average Precision at an Intersection over Union threshold of 0.5 (mAP@0.5). The results, illustrated in Fig. 9, show that all models perform well, with scores consistently above 80% across all metrics.

Among the models, YOLOv8n achieved the highest performance, with a mean precision of 92.99%, a mean recall of 93.47%, and an *mAP*@0.5 of 92.5%. This indicates that YOLOv8n not only detects objects with high precision but also achieves excellent recall rates, making it one of the most reliable choices for applications requiring high precision and robustness.

The results highlight the incremental improvements introduced in the newer YOLO versions, with YOLOv8n emerging as the best-performing model for detecting inattentive driving behaviors. These performance metrics affirm that YOLOv8n is particularly suited for tasks that demand high precision and recall. However, YOLOv7, YOLOv6n, and YOLOv5n remain effective alternatives, especially in resource-constrained environments, where slightly reduced computational demands may be prioritized over marginal gains in accuracy and precision.

The performance analysis presented in Fig. 10 reveals YOLOv8n mAP@50% of 0.925, demonstrating notable precision in detecting "Driver Looking away" (0.992) and "Eyes Open" (0.956). While the model maintained above 80% performance across most classes, its mAP@0.5

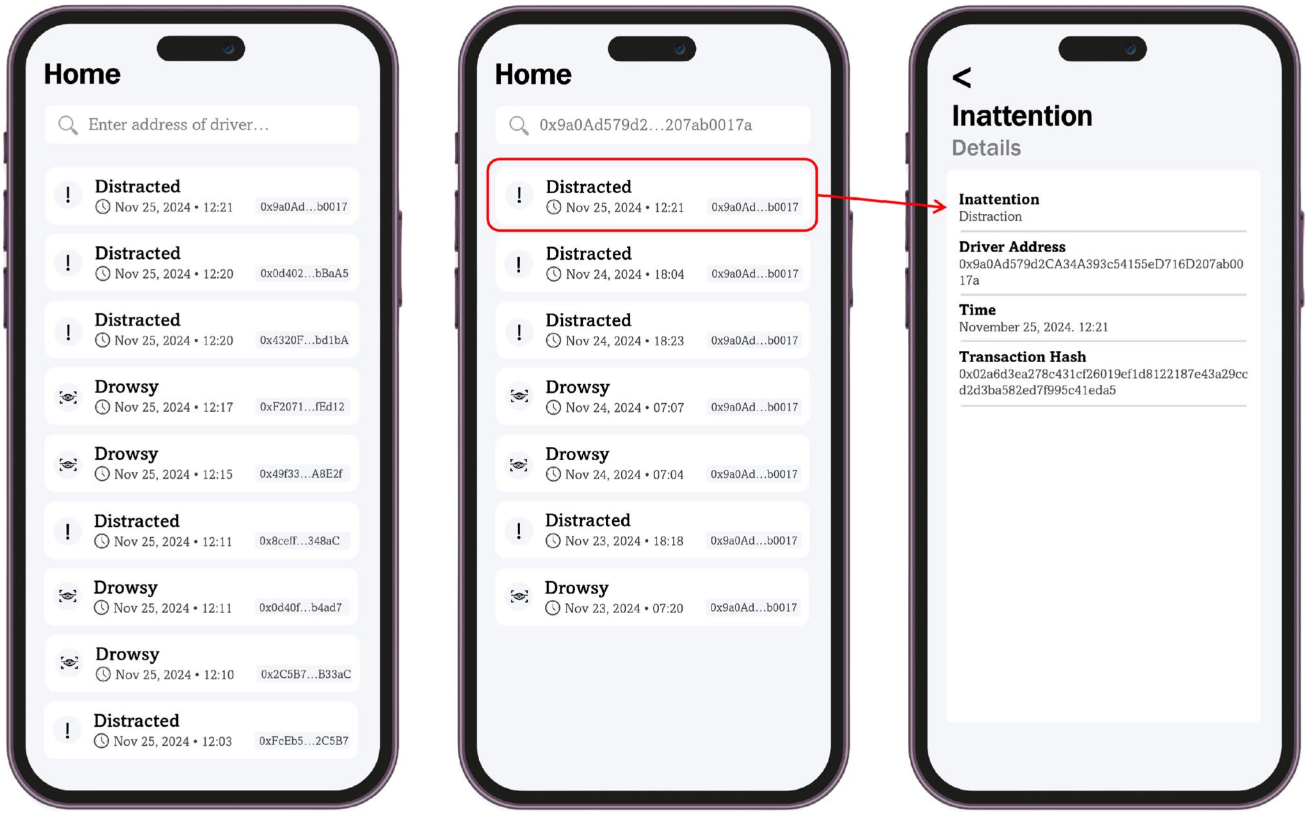

Fig. 7 The DApp's main interface displays a comprehensive list of recorded inattention incidents. Stakeholders can search for specific driver records by entering the driver's blockchain address. Tapping any record reveals detailed event information, including timestamps, transaction hashes, and classification of the inattentive behavior

dropped when detecting "Yawning" (0.787). As evidenced in Fig. 11, YOLOv7 achieved a mAP@50% of 0.909, excelling in the detection of "Driver Looking away" (0.991) and "Driver" (0.985) categories, despite its relatively weaker performance in identifying "Yawning" (0.721). The results documented in Fig. 12 indicate that YOLOv6n recorded the lowest performance with a mAP@50% of 0.831. Although this model maintained reasonable precision in detecting "Driver Looking away" (0.984) and "Driver talking on phone" (0.916), it faced challenges with "Yawning" (0.579) and "Texting" (0.742). Figure 13 demonstrates YOLOv5n good performance, achieving a mAP@50% of 0.931 and exhibiting good detection capabilities across some classes, particularly for "Driver Looking away" (0.993) and "Driver talking on phone" (0.966).

The performance of YOLO models (YOLOv8n, YOLOv7, YOLOv6n, and YOLOv5n) is analyzed based on metrics such as precision, recall, F1-score, accuracy, and mAP@0.5, as illustrated in Table 4: YOLO Comparison Table and Fig. 9: YOLO Models Comparison. These metrics are evaluated across various inattentive driving behaviors, including drinking, texting, yawning, and more. Among the models, YOLOv8n achieves the best performance, with a precision of 92.99%, recall of 93.47%, and F1-score of 92.47%, making it the most reliable for detecting driver inattention. YOLOv7 and YOLOv6n perform comparably, maintaining an average accuracy above 90%, while YOLOv5n shows slightly lower but still competitive performance, with an average accuracy of 89.94%. As shown in Fig. 9, YOLOv8n consistently excels across all metrics, confirming its robustness for real-time object detection in driver monitoring systems. These results highlight the incremental advancements in newer YOLO versions, underscoring YOLOv8n's suitability for high-precision safety applications.

Figure 15 presents representative inference results obtained using the trained YOLOv8n model on unseen test images. The figure demonstrates the model's ability to perform simultaneous multi-object detection by identifying several driver-related classes within a single frame, including Driver Looking Away, Eyes Open or Eyes Closed, and Seat Belt. Each detected object is enclosed by a bounding box and assigned a confidence score, illustrating the model's confidence in its predictions.Across the test samples, the YOLOv8n model consistently detects and localizes multiple driver-related objects and behaviors within the same image, including Driver Looking Away, Eyes Open or Eyes Closed, and Seat Belt. The ability to simultaneously identify

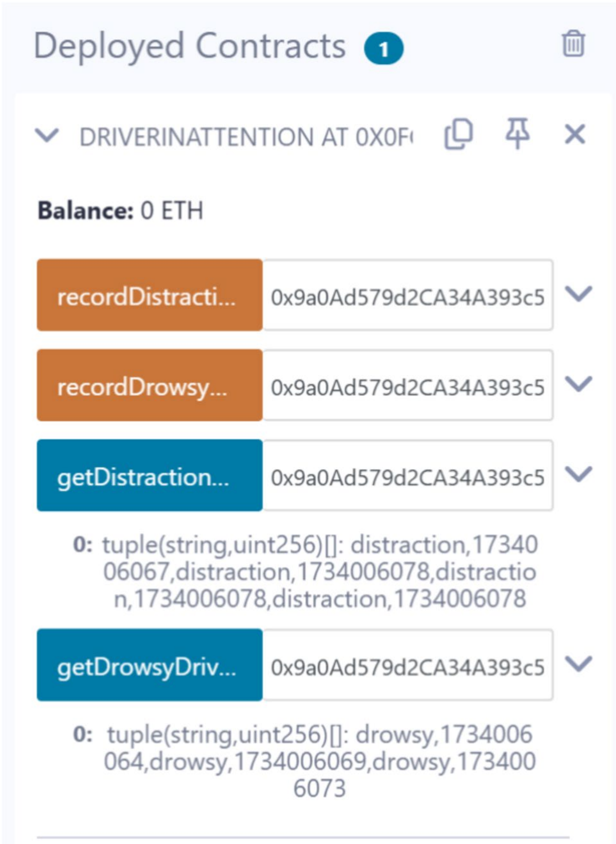

Fig. 8 The image illustrates a smart contract designed to track driver inattention. When a specific driver (identified by their blockchain address) is detected being inattentive, the *recordDrowsyDriver* function is triggered. The contract logs instances of drowsiness, such as closed eyes, and in this example, it records that the driver's eyes were closed twice

these inattention-related classes from a single camera image demonstrates the effectiveness of the proposed multi-class object detection approach for comprehensive driver inattention monitoring.

7.3 Detection-centric evaluation and per-class performance analysis

To ensure methodological correctness for object detection evaluation, this study emphasizes detection-specific metrics rather than overall accuracy. Metrics such as **mean Average Precision (mAP@0.5)** (Table 5) and **per-class Average Precision (AP)** (Table 6) provide a more reliable assessment of both localization and classification performance of the YOLO models, especially in the presence of class imbalance across driver behavior categories.

Certain behavior classes such as *yawning* and *seat belt detection* contain fewer representative samples and exhibit higher occlusion patterns, which directly influence their AP values. This highlights the effect of class imbalance and real-world visual complexity on detection performance. Therefore, model comparison and selection in this work are based primarily on AP and mAP metrics.

The results show that although YOLOv5n achieves the highest overall mAP, YOLOv8n provides the most balanced and consistent performance across most important driver behavior classes. This balanced performance is critical for real-time in-vehicle monitoring, where robustness across diverse behaviors is more important than peak performance in a few categories.

Certain behavior classes such as yawning and seat belt detection contain fewer representative samples and exhibit higher occlusion patterns, which directly influence their AP values. This highlights the effect of class imbalance and real-world visual complexity on detection performance. Therefore, model comparison and selection in this work are based primarily on AP and mAP metrics.

7.4 Edge device inference performance

The feasibility of real-time deployment was evaluated on a Raspberry Pi 5 (8 GB RAM) during continuous YOLOv8n inference. The system achieved 4.05 FPS with a per-frame latency of 246.60 ms, while maintaining moderate resource utilization (41.70% CPU and 24.60% RAM), as summarized in Table 7. The trained YOLOv8n model weight occupies only 6100 KB (approximately 6.1 MB), demonstrating its suitability for practical deployment on low-power edge devices.

Although the FPS is lower than GPU-based systems, it is sufficient for the proposed framework due to the 3-second temporal verification rule. At this rate, approximately 12 consecutive frames are analyzed before confirming driver inattention, which improves robustness and minimizes false alarms. The measured latency is also comparable to human reaction time (250–300 ms), indicating suitability for real-time monitoring.

The device was powered using the official 27 W USB-C supply. However, given the moderate CPU and memory usage observed, the practical operating power during inference is expected to remain well below this limit, confirming that the system is power-efficient and appropriate for continuous vehicular embedded deployment.

7.5 Blockchain network performance

The performance evaluation of the IBFT 2.0 Proof of Authority (POA) blockchain network revealed distinct patterns across different operational parameters, as documented in Table 8. To evaluate the network's scalability, Hyperledger Caliper was employed as the load generation tool. The analysis focused on three key operations under varying

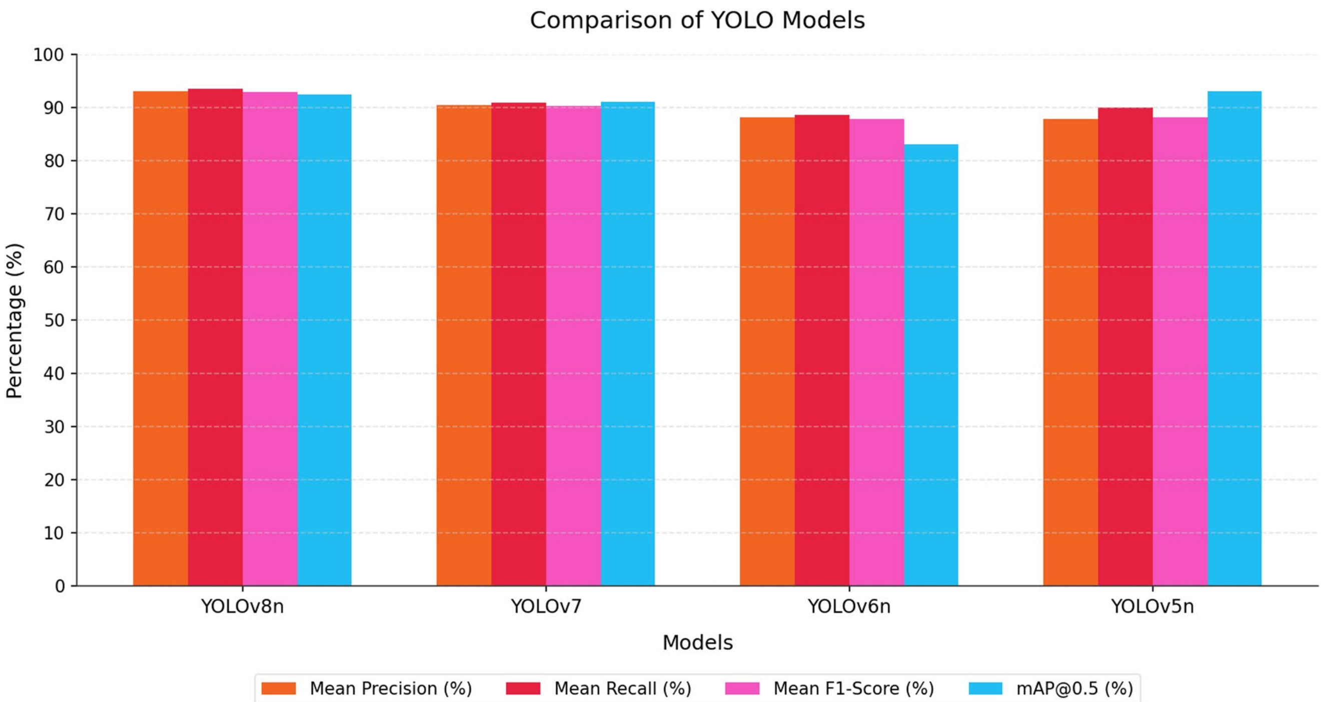

Fig. 9 YOLO (YOLOv8n, YOLOv7, YOLOv6n & YOLOv5n) Models Comparison

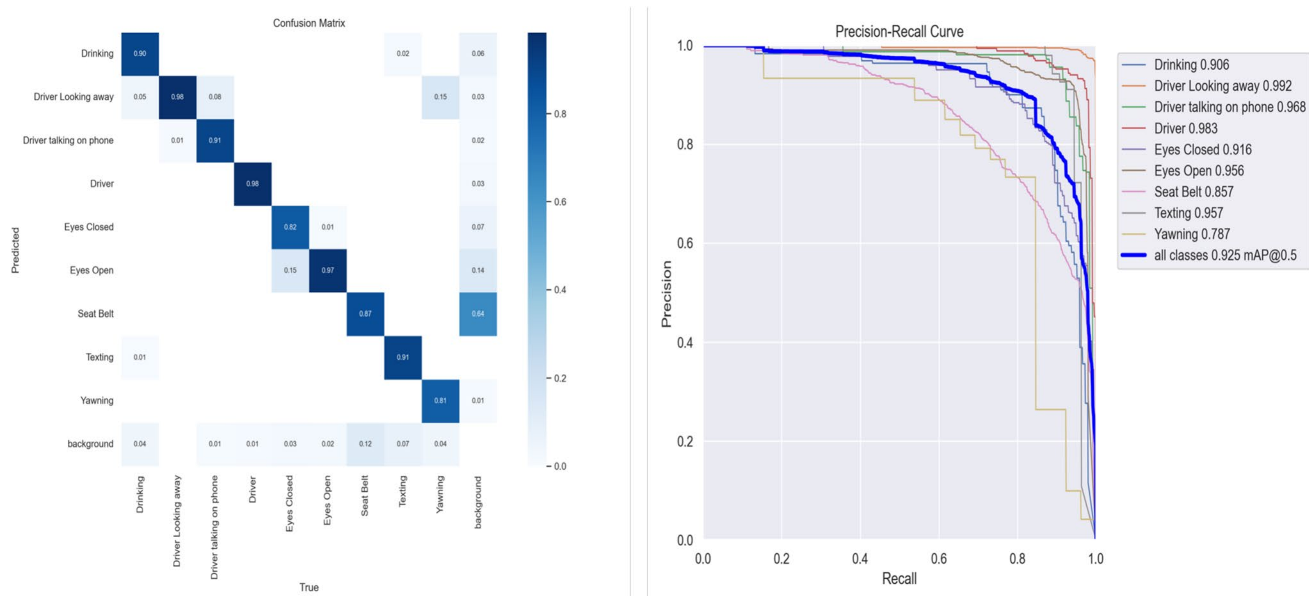

Fig. 10 YOLOv8n Confusion Matrix and Precision-Recall Curve showing the mAP@0.5 IoU

send rates ranging from 100 to 1000 requests per second: *open* (creating a new instance on the ledger), *transfer* (state-updating transactions such as `recordDrowsyDriver()` or `recordDistractionDriver()`), and *query* (read-only calls such as `getDrowsyDriverEvents()`). Performance evaluation was conducted using Hyperledger Caliper (v0.6.0) as the load generation tool. All reported metrics represent the arithmetic mean of ten independent benchmark rounds. Run-to-run variance was low across all rounds, consistent with the deterministic block-finalisation behaviour of IBFT 2.0 on a stable local network. The block period was configured to 1 second and maintained constant across all experimental conditions (Fig. 14).

The network demonstrated a robust stability and perfect reliability with a 100% success rate across all send rates. The latency characteristics, as illustrated in Fig. 15 (left),

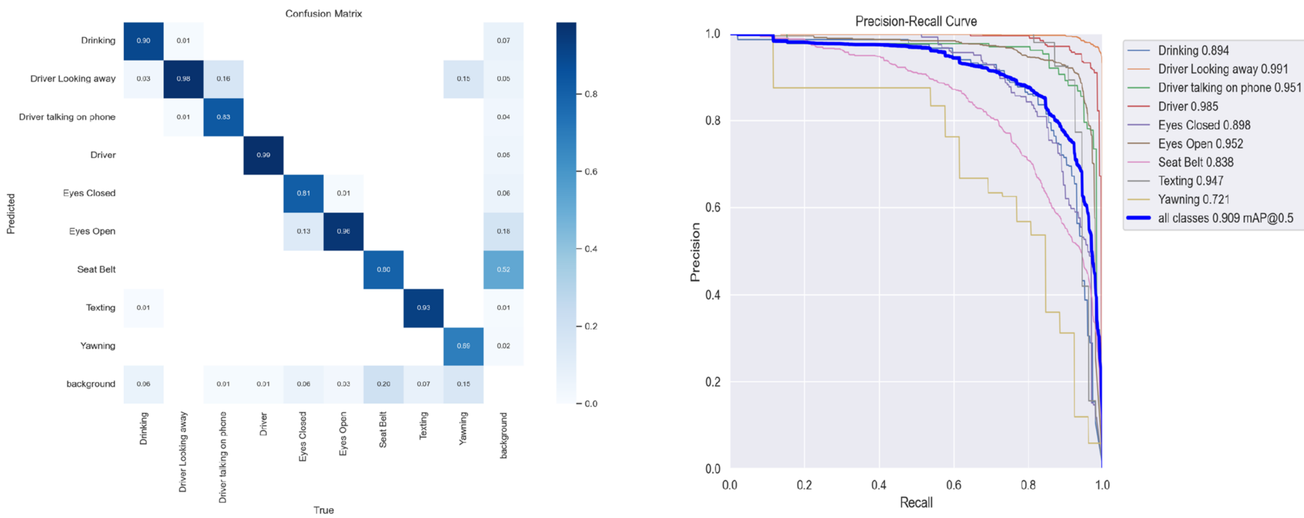

Fig. 11 YOLOv7 Confusion Matrix and Precision-Recall Curve showing the mAP@0.5 IoU

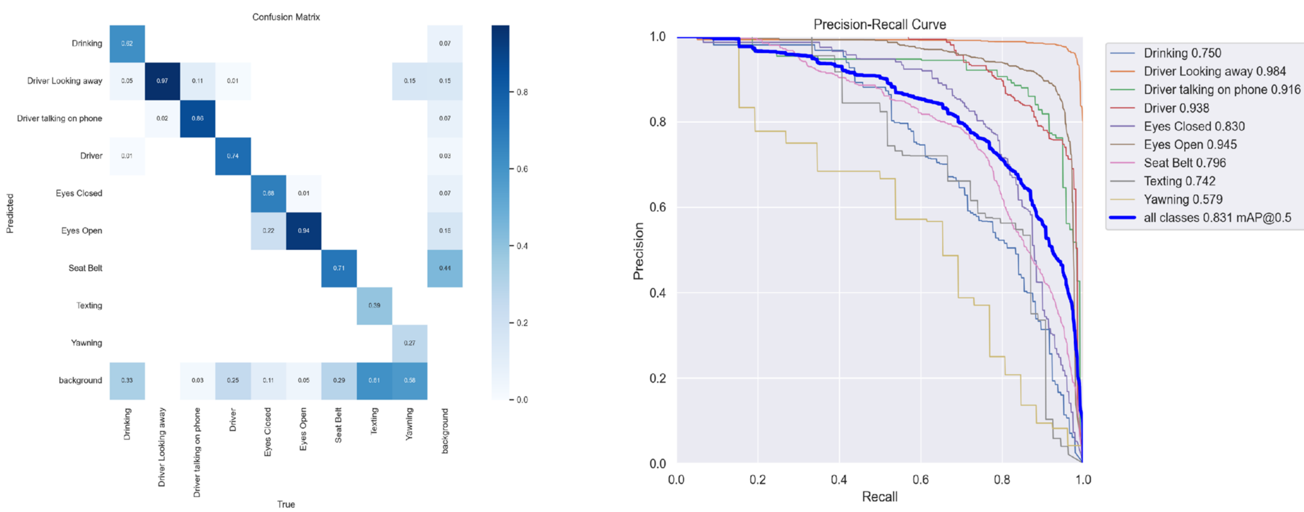

Fig. 12 YOLOv6n Confusion Matrix and Precision-Recall Curve showing the mAP@0.5 IoU

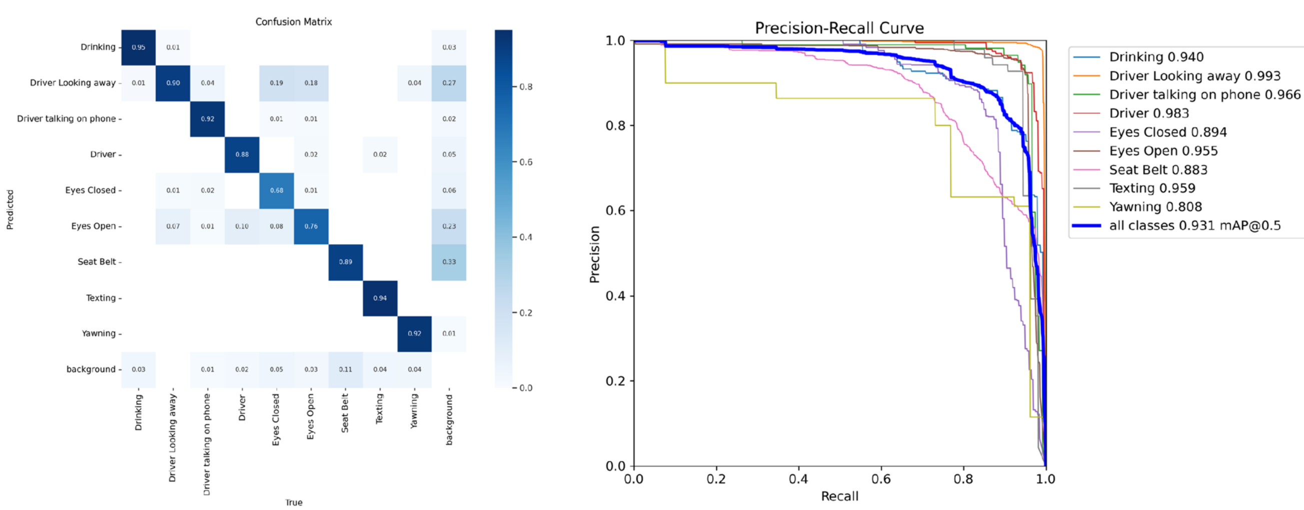

Fig. 13 YOLOv5n Confusion Matrix and Precision-Recall Curve showing the mAP@0.5 IoU

Table 4 YOLO Comparison Table

Models	YOLOv8n			YOLOv7			YOLOv6n			YOLOv5n		
	Precision (%)	Recall (%)	F1-Score (%)	Precision (%)	Recall (%)	F1-Score (%)	Precision (%)	Recall (%)	F1-Score (%)	Precision (%)	Recall (%)	F1-Score (%)
Drinking	93.75	91.84	92.78	95.74	91.84	93.75	88.57	80.52	84.35	97.94	96.94	97.44
Driver Looking Away	97.03	88.29	92.45	84.48	84.48	84.48	82.91	82.91	82.91	93.75	55.90	70.04
Driver Talking on Phone	84.26	98.91	91.00	75.45	83.00	79.05	77.48	86.87	81.90	79.31	94.85	86.38
Driver	98.99	100.00	99.49	99.00	100.00	99.50	97.37	97.37	97.37	81.48	97.78	88.89
Eyes Closed	84.54	86.32	85.42	86.17	85.26	85.71	75.56	83.95	79.53	80.95	85.00	82.93
Eyes Open	98.98	76.98	86.61	98.97	75.59	85.71	98.95	71.21	82.82	73.08	80.00	76.38
Seat Belt	80.56	100.00	89.23	76.92	100.00	86.96	75.53	100.00	86.06	84.76	100.00	91.75
Texting	100.00	98.91	99.45	100.00	97.89	98.94	100.00	95.12	97.50	100.00	98.95	99.47
Yawning	98.78	100.00	99.39	97.18	100.00	98.57	96.83	100.00	98.39	98.92	100.00	99.46
Average	92.99	93.47	92.87	90.43	90.90	90.30	88.13	88.66	87.87	87.80	89.94	88.08

Table 5 Overall mAP@0.5 Comparison

Model	mAP@0.5
YOLOv5n	0.931
YOLOv6n	0.831
YOLOv7	0.909
YOLOv8n	0.925

Table 6 Per-Class Average Precision (AP) Comparison Across YOLO Variants

Behavior Class	YOLOv5n AP	YOLOv6n AP	YOLOv7 AP	YOLOv8n AP
Drinking	0.940	0.750	0.894	0.906
Driver looking away	0.993	0.984	0.991	0.992
Driver talking on phone	0.966	0.916	0.951	0.968
Driver (face/body)	0.983	0.938	0.985	0.983
Eyes closed	0.894	0.830	0.898	0.916
Eyes open	0.955	0.945	0.952	0.956
Seat belt	0.883	0.796	0.838	0.857
Texting	0.959	0.742	0.947	0.957
Yawning	0.808	0.579	0.721	0.787

Table 7 Edge Device Performance Summary on Raspberry Pi 5 (8GB RAM)

Metric	Measured Value	Deployment Implication
Frames Per Second (FPS)	4.05 FPS	Approximately 12 frames are analyzed over the 3-second verification window, enabling reliable behavioral confirmation and reducing false alarms.
Inference Latency	246.60 ms	Comparable to human reaction time (250–300 ms), supporting real-time driver monitoring.
CPU Utilization	41.70%	Indicates significant processing headroom for stable and continuous operation.
RAM Utilization	24.60%	Demonstrates memory-efficient execution suitable for long-term deployment.
Power Supply Capacity	27 W (official USB-C PSU)	Provides sufficient headroom for peak load conditions and connected peripherals.
Expected Operating Power	<27 W	Practical power consumption during inference remains well below the hardware design limit.

showed notable variations among different operations. Open operations exhibited the highest latency, peaking at 4.43 seconds at 800 Req/s before decreasing to 3.6 seconds at 1000 Req/s. Transfer operations displayed a steady increase in latency from 0.84 to 3.24 seconds as the send rate increased. In contrast, query operations maintained a consistent low latency of 0.001 seconds regardless of the send rate.

The throughput analysis, depicted in Fig. 15 (right), revealed distinct performance characteristics for each

Table 8 IBFT 2.0 PoA Blockchain Network Performance

Send Rate	Success	Failure	Avg Latency (s)			Throughput (TPS)		
(Req/s)			Open	Query	Transfer	Open	Query	Transfer
100	100.000	0.000	1.040	0.001	0.840	92.000	100.100	94.000
200	200.000	0.000	3.090	0.001	1.730	96.000	200.600	128.500
400	400.000	0.000	4.150	0.001	2.990	94.300	291.100	121.300
600	600.000	0.000	4.080	0.001	3.000	95.200	271.100	125.600
800	800.000	0.000	4.430	0.001	3.040	94.400	296.000	121.800
1000	1000.000	0.000	3.600	0.001	3.240	105.100	299.100	119.600

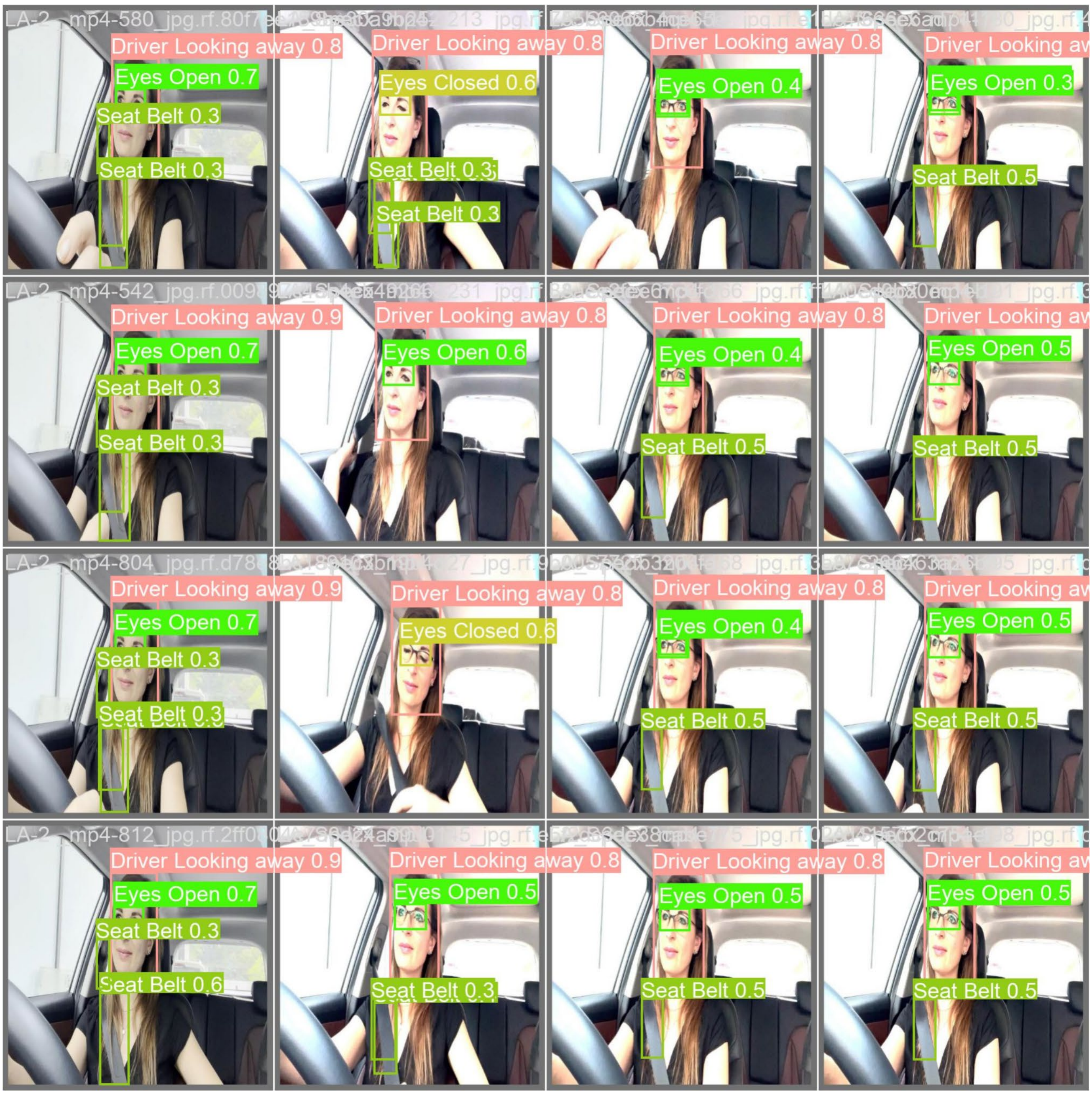

Fig. 14 YOLOv8n inference result using test dataset

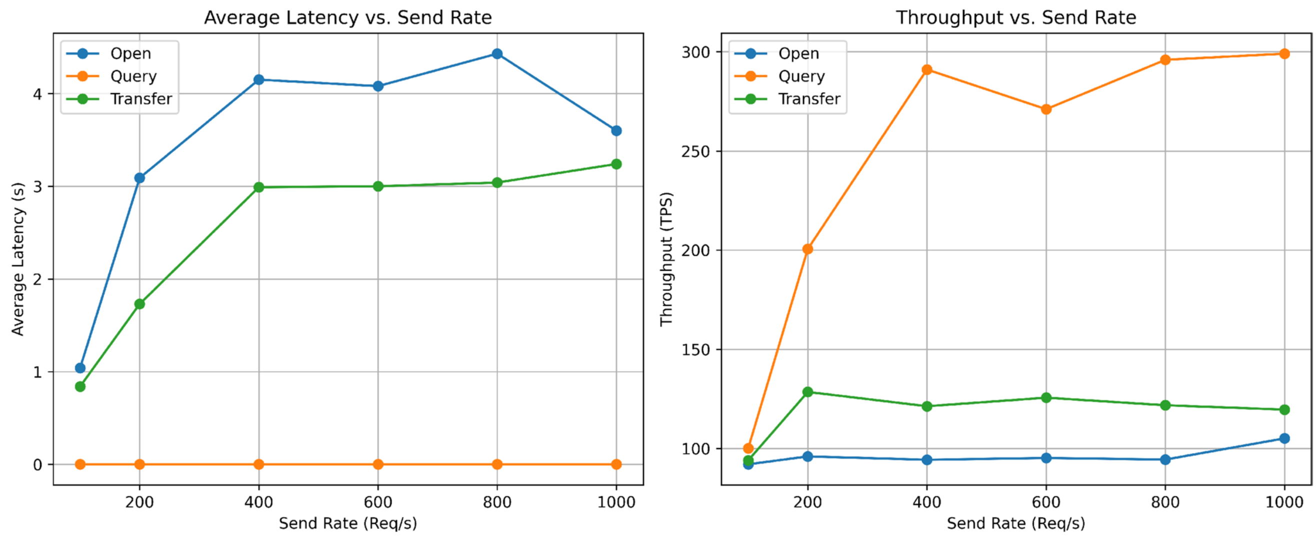

Fig. 15 IBFT 2.0 PoA blockchain network performance

operation type. Query operations showed the most impressive scalability, with throughput increasing significantly from 100.1 TPS at 100 Req/s to approximately 299.1 TPS at 1000 Req/s. Open operations maintained relatively stable throughput around 95 TPS across most send rates, with a slight improvement to 105.1 TPS at 1000 Req/s. Transfer operations initially increased from 94 TPS to 128.5 TPS at 200 Req/s, before stabilizing around 120 TPS for higher send rates.

The observed latency values are consistent with the decomposition in Eq. 1. The 0.84-second baseline transfer latency corresponds to $T_{\text{consensus}}$ approaching the 1-second block period at low load with co-located validators. Increases at higher send rates (up to 3.24 s) reflect block-filling effects as the per-block gas capacity is approached, not a fundamental protocol constraint.

7.6 Ablation study and performance analysis

To evaluate the individual contributions of the core components within the proposed framework, an ablation study was conducted focusing on the Temporal Verification (TV) logic and the Blockchain (BC) logging module. As illustrated in Table 9, maintaining a consistent mAP@0.5 of 92.5% across all configurations confirms the robustness of the underlying YOLOv8n detection engine. However, the study reveals that the TV logic is the critical determinant for system reliability; removing it results in the False Alarm Rate (FAR) climbing from 1.1% to 7.9%, as the system loses the ability to filter out transient natural movements like standard blinking. Regarding system efficiency, the processing latency at the edge remains low (≈246.60 ms), while the inclusion of blockchain introduces a measurable network latency of 840.0 ms. Crucially, since the blockchain operations are handled asynchronously, they do not interfere with the real-time processing required for safety alerts. The full system configuration thus emerges as the most effective, providing high precision and a minimized False Alarm Rate alongside immutable data security without compromising immediate driver safety interventions.

Table 9 Ablation study of the proposed integrative framework

Configuration	mAP@0.5 (%)	FAR (%)	Proc. Latency (ms)	Net. Latency (ms)
Full System	**92.5**	**1.1**	**246.60**	**840.0**
–Blockchain	92.5	1.1	246.60	0.00
–Temporal Logic	92.5	7.9	243.60	840.0
YOLOv8n only	92.5	7.9	246.60	0.00

7.7 Comparison with existing technologies

A comparative analysis of the proposed solution against existing technologies was conducted, focusing on three critical parameters: simultaneous inattention and eye-state detection using a single camera, decentralized storage capabilities through blockchain integration, and real-time inattention alerting mechanisms. Table 10 presents a detailed comparison of these technological features across various studies.

While previous studies such as [3, 7, 24] successfully implemented real-time inattention alerts, they faced limitations in two crucial aspects. First, these studies required multiple cameras or separate systems to detect inattention and eye-state conditions, increasing system complexity and implementation costs. Second, they relied on traditional centralized storage systems, which may present vulnerabilities in terms of data security and integrity.

The developed framework addresses these limitations through several technological innovations. The implementation

Table 10 Comparison of studies on driver inattention detection

Studies	Distraction	Drowsiness	Decentralized Access	Real-time Alert
Sinha et al. [22]	×	✓	×	✓
Soman et al. [24]	×	✓	×	✓
Ziryawulawo et al. [30]	×	✓	×	✓
Phan et al. [21]	×	✓	×	✓
Ganguly et al. [11]	×	✓	×	✓
Shakeel et al. [23]	×	✓	×	✓
Christy et al. [6]	✓	×	×	×
Hossain et al. [12]	✓	×	×	✓
Bekka et al. [3]	✓	×	×	✓
Proposed Tech.	✓	✓	✓	✓

of advanced computer vision algorithms enables simultaneous detection of both inattention and eye-state conditions using a single camera setup, resulting in reduced system complexity and cost. Blockchain integration provides secure, immutable, and decentralized storage capabilities, ensuring data integrity and transparent access control. The system architecture maintains real-time alerting capabilities while enhancing functionality through a unified detection approach. Based on the comparative analysis, the proposed solution effectively addresses the key technological gaps identified in existing systems, delivering an integrated, efficient, and secure framework for driver monitoring and safety enhancement.

8 Ethical and privacy considerations

The deployment of a system that records and distributes driver behavior data to institutional stakeholders necessitates a rigorous treatment of privacy, consent, and regulatory compliance. The proposed system is designed around a Privacy-by-Design principle that addresses these concerns at the architectural level rather than as an afterthought.

Edge-only processing and data minimization Raw visual data—camera frames and facial imagery—are processed entirely on the Raspberry Pi at the edge and are never transmitted to the blockchain or any external server. Only behavioral event classifications (e.g., drowsiness, distraction) and a pseudonymous blockchain address are recorded on-chain. Recording is event-triggered and fires only when inattention persists beyond the confirmed threshold (three seconds), not as a continuous stream. Such design ensures the system acts strictly as an event logger, not a surveillance tool.

Driver consent and data governance In an institutional deployment context (e.g., a logistics fleet or a road safety authority program), drivers are informed of monitoring conditions as part of their employment or participation agreement, consistent with occupational monitoring frameworks recognized under GDPR and comparable national regulations. Stakeholder access to a driver's records is not granted by default: each requesting organization must first receive explicit approval from the driver before the smart contract permits data retrieval. No personally identifiable information is stored on-chain; the mapping between a blockchain address and a physical identity is maintained off-chain by the deploying institution under its own data controller obligations.

Reconciling immutability with the right to erasure Blockchain immutability and the right to erasure under data protection regulations are not inherently contradictory when the stored data is pseudonymous. Should a driver invoke the right to erasure, the institution can sever the off-chain identity link, rendering the on-chain records effectively anonymous without altering the blockchain state. Because raw visual data resides only on the edge device and is never committed to the chain, physical deletion of that local data is straightforward and complete.

Access control and misuse prevention Unauthorized data mining is prevented at the smart contract level through the RBAC registry described in Section 3.3. Only registered, authenticated stakeholder addresses can invoke retrieval functions, and the contract enforces role boundaries in code rather than in policy—no party can escalate its own privileges without an explicit governance transaction. Every query invocation is recorded as an immutable Solidity `event`, creating an auditable trail of who accessed which records and when, deterring surveillance misuse and providing accountability to the driver.

9 Limitations

While the proposed framework demonstrates the viability of integrating edge AI with permissioned blockchain for driver safety monitoring, several limitations define the scope of the current work and inform future research directions.

Blockchain validator set size and liveness resilience The experimental network operates with $n = 4$ validator nodes, the minimum configuration for IBFT 2.0 to provide any Byzantine fault tolerance ($f = \lfloor (n-1)/3 \rfloor = 1$). While this size aligns naturally with the four-stakeholder governance model, it implies a strict liveness boundary. The network halts if two validators are simultaneously unavailable, as the remaining two cannot form the commit quorum of three. The safety property, which guarantees that no conflicting record is ever finalised, is maintained as long as fewer than two nodes are Byzantine. Expanding the validator set beyond the minimum is identified as a direction for future work to improve this liveness boundary.

Co-located experimental deployment The blockchain performance results in Table 8 were obtained with all four validator nodes running as instances on a single physical machine. This configuration deliberately isolates the software-level throughput characteristics of IBFT 2.0 by eliminating inter-node network variance. The reported consensus latency values (0.84–4.43 s) therefore represent a lower bound; in a geographically distributed deployment, inter-node round-trip time will increase $T_{\text{consensus}}$ accordingly. Similarly, the Raspberry Pi communicated with the Besu node over the local network during experiments; the effect of network variability on T_{network} under real-world deployment conditions was not independently characterised.

Consensus protocol alignment IBFT 2.0 was selected at the time of system design for its well-established deterministic finality properties and its documented behaviour at small validator sets. Hyperledger Besu also supports QBFT as an alternative Proof of Authority consensus protocol. Evaluation of QBFT as a drop-in replacement, and comparison of its round-change performance at $n = 4$ under failure scenarios, is identified as a direction for future work.

Connectivity resilience The current implementation assumes eventual network connectivity for blockchain event submission. While the asynchronous local queue prevents data loss during brief outages, behaviour under prolonged disconnection, including scenarios involving queue capacity limits and clock-drift effects on event timestamps, was not evaluated. Evaluation of behaviour under prolonged disconnection, including queue capacity management and timestamp correction strategies, is identified as an avenue for future work.

10 Conclusion

This research successfully integrates deep learning and blockchain technologies to create a robust system for detecting and managing driver inattention. By leveraging YOLOv8n, which achieved the highest performance metrics with a mean precision of 92.99%, mean recall of 93.47%, and mAP@0.5 of 92.5%, the system ensures reliable real-time detection of inattentive behaviors. The deployment of Hyperledger Besu with IBFT 2.0 consensus further enhances the system's data management and scalability, with a throughput of up to 128.5 transactions per second (TPS) and latency ranging from 0.84 seconds to 4.43 seconds, ensuring secure and efficient data storage.

The implementation of DApps allows role-based access for stakeholders, such as road safety officials and insurance companies, enabling transparency and secure retrieval of driver behavior records. This integrated approach reduces system complexity through a single-camera setup for simultaneous inattention and eye-state detection, while also addressing the susceptibility to data tampering that characterises traditional centralized systems. Overall, the proposed solution demonstrates the potential of combining advanced deep learning techniques with blockchain technology to enhance road safety through real-time monitoring, reliable data management, and stakeholder accessibility. Although the proposed system demonstrates strong detection performance under the experimental conditions, extensive real-world validation across diverse drivers, vehicles, lighting conditions, and camera configurations remains an important direction for future research to further establish practical deployment robustness. Future work could also explore additional use cases, optimize blockchain performance, and expand detection capabilities to accommodate a broader range of inattentive behaviors. Future research will focus on enhancing the statistical robustness of the experimental evaluation by incorporating k-fold cross-validation, confidence interval estimation, and variance analysis across multiple training runs. These additions will enable more rigorous performance differentiation among YOLO variants and improve the reliability of model selection for safety-critical applications. Furthermore, extended real-world validation under diverse driving conditions and larger-scale datasets will be conducted to strengthen generalizability. On the system side, future work will also explore optimization techniques such as model pruning, quantization, and hardware acceleration to further improve inference speed, energy efficiency, and scalability for embedded deployment.

Author Contributions Odinachi Udemezuo Nwankwo: contributed to the research methodology, system design, artificial intelligence simulation, manuscript writing, and editing. Muhammad Rasyid Redha Ansori: contributed to the smart contract design, manuscript writing, and editing. Gifar Arif Haryadi: contributed to the blockchain network simulation, manuscript writing, and editing. Dong-Seong Kim: secured funding and provided supervision. Jae-Min Lee: secured funding and provided supervision. All authors reviewed and approved the final manuscript.

Funding This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the IITP grant funded by the Korea government (MSIT) (IITP-2026-RS-2020-II201612, 20%) and by Priority Research Centers Program through the NRF funded by the MEST (2018R1A6A1A03024003, 20%) and bythe MSIT, Korea, under the ITRC support program (IITP-2026-RS-2024-00438430,20%) and by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2025-25431637, 20%) and the regional innovation system & education (RISE)-(Idea Start-up Valley) program through the Gyeongbuk RISE Center, funded by the Ministry of Education (MOE) and the Gyeongsangbuk-do, Republic of Korea (2026-rise-15-105, 20%).

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Aditya Y, Jain P, Tiwari R (2025) Improved health care monitoring system using proposed LRRN-DLNN algorithm based on CPPFFR AI data preservation and ESRCC-based security. Iranian J Sci Tech Trans Electr Eng 49(2). https://doi.org/10.1007/s40998-025-00858-7
2. Adochiei I-R, Știrbu O-I, Adochiei N-I, Pericle-Gabriel M, Larco C-M, Mustata S-M, Costin D (2020) Drivers' drowsiness detection and warning systems for critical infrastructures. In: 2020 International Conference on e-Health and Bioengineering (EHB), pp 1–4. https://doi.org/10.1109/EHB50910.2020.9280165
3. Bekka R, Kherbouche S, Houda EB (2022) Distraction detection to predict vehicle crashes: A deep learning approach. Computación y Sistemas 26. https://doi.org/10.13053/cys-26-1-3871
4. Brodbeck V, Kuhn A, von Wegner F, Morzelewski A, Tagliazucchi E, Borisov S, Michel CM, Laufs H (2012) Eeg microstates of wakefulness and nrem sleep. Neuroimage 62(3):2129–2139. https://doi.org/10.1016/j.neuroimage.2012.05.060
5. Chang W-J, Chen L-B, Chiou Y-Z (2018) Design and implementation of a drowsiness-fatigue-detection system based on wearable smart glasses to increase road safety. IEEE Trans Consum Electron 64(4):461–469. https://doi.org/10.1109/TCE.2018.2872162
6. Christy A, Shyry P, Meeragandhi G, Praveena MD (2021) Driver distraction detection and early prediction and avoidance of accidents using convolutional neural networks. J Phys: Conf Ser 1770:012007. https://doi.org/10.1088/1742-6596/1770/1/012007
7. Chang RC-H, Wang C-Y, Chen W-T, Chiu C-D (2022) Drowsiness detection system based on perclos and facial physiological signal. Sensors 22(14)
8. Choi G, Ziyang G, Wu J, Esposito C, Choi C (2023) Multi-modal biometrics based implicit driver identification system using multi-tf images of ecg and emg. Comput Biol Med 159:106851. https://doi.org/10.1016/j.compbiomed.2023.106851
9. Du M, Chen Q, Chen J, Ma X (2021) An optimized consortium blockchain for medical information sharing. IEEE Trans Eng Manage 68(6):1677–1689. https://doi.org/10.1109/TEM.2020.2966832
10. Fan C, Lin C, Khazaei H, Musilek P (2022) Performance analysis of hyperledger besu in private blockchain. In: 2022 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), pp 64–73. https://doi.org/10.1109/DAPPS55202.2022.00016
11. Ganguly B, Dey D, Munshi S (2022) An integrated system for drivers' drowsiness detection using deep learning frameworks. In: 2022 IEEE VLSI Device Circuit and System (VLSI DCS), pp 55–59. https://doi.org/10.1109/VLSIDCS53788.2022.9811442
12. Hossain MU, Rahman MA, Islam MM, Akhter A, Uddin MA, Paul BK (2022) Automatic driver distraction detection using deep convolutional neural networks. Intell Syst Appl 14:200075. https://doi.org/10.1016/j.iswa.2022.200075
13. Jacobé de Naurois C, Bourdin C, Stratulat A, Diaz E, Vercher J-L (2019) Detection and prediction of driver drowsiness using artificial neural network models. Accident Analysis & Prevention 126:95–104. https://doi.org/10.1016/j.aap.2017.11.038. In: 10th International Conference on Managing Fatigue: Managing Fatigue to Improve Safety, Wellness, and Effectiveness"
14. Khatoon A (2020) A blockchain-based smart contract system for healthcare management. Electronics 9(1). https://doi.org/10.3390/electronics9010094
15. Kukkala VK, Tunnell J, Pasricha S, Bradley T (2018) Advanced driver-assistance systems: A path toward autonomous vehicles. IEEE Consumer Electronics Magazine 7(5):18–25. https://doi.org/10.1109/MCE.2018.2828440
16. Kalisetti V, Vasarla V, Kolli SB, Varaparla R, Enireddy V, Mohammed M (2023) Analysis of driver drowsiness detection methods. In: 2023 Second International Conference on Electronics and Renewable Systems (ICEARS), pp 1481–1485. https://doi.org/10.1109/ICEARS56392.2023.10084986
17. Li G, Chung W-Y (2013) Detection of driver drowsiness using wavelet analysis of heart rate variability and a support vector machine classifier. Sensors 13(12):16494–16511
18. MTech (2023) Driver Distraction - Single Camera Dataset. Roboflow. visited on 2024-02-27. https://universe.roboflow.com/mtech-yi2na/driver-distraction_single-camera
19. Nandhini PS, Kuppuswami S, Malliga S, Srinath P, Veeramanikandan P (2022) Driver drowsiness detection using deep learning. In: 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), pp 1031–1036. https://doi.org/10.1109/ICCMC53470.2022.9754053
20. Patel V (2019) A framework for secure and decentralized sharing of medical imaging data via blockchain consensus. Health Informatics J 25(4):1398–1411. https://doi.org/10.1109/TNSM.2023.3241610
21. Phan A-C, Nguyen N-H-Q, Trieu T-N, Phan T-C (2021) An efficient approach for detecting driver drowsiness based on deep learning. Appl Sci 11:8441. https://doi.org/10.3390/app11188441
22. Sinha A, Aneesh RP, Gopal SK (2021) Drowsiness detection system using deep learning. In: 2021 7th International Conference on Bio Signals, Images, and Instrumentation (ICBSII), pp 1–6. https://doi.org/10.1109/ICBSII51839.2021.9445132
23. Shakeel MF, Bajwa NA, Anwaar AM, Sohail A, Khan A, Khan H-U-R (2019) Detecting driver drowsiness in real time through deep learning based object detection. In: International Work-Conference on Artificial and Natural Neural Networks
24. Soman SP, Senthil Kumar G, K M A (2023) Internet-of-things-assisted artificial intelligence-enabled drowsiness detection framework. IEEE Sensors Letters 7(7):1–4. https://doi.org/10.1109/LSENS.2023.3289143
25. Usman M, Qamar U (2020) Secure electronic medical records storage and sharing using blockchain technology. Procedia Computer Science. In: 2019 International Conference on Identification, Information and Knowledge in the Internet of Things 174:321–327. https://doi.org/10.1016/j.procs.2020.06.093
26. Wang F, Wan Y, Li M, Huang H, Li L, Hou X, Pan J, Wen Z, Li J (2023) Recent advances in fatigue detection algorithm based on eeg. Intell Autom Soft Comput 35(3):3573–3586
27. Wang X, Xu R, Zhang S, Zhuang Y, Wang Y (2022) Driver distraction detection based on vehicle dynamics using naturalistic driving data. Transportation Research Part C Emerging Technologies 136:103561. https://doi.org/10.1016/j.trc.2022.103561

28. Xiao W, Liu H, Ma Z, Chen W (2022) Attention-based deep neural network for driver behavior recognition. Futur Gener Comput Syst 132:152–161. https://doi.org/10.1016/j.future.2022.02.007
29. Yan Y, Zhong S, Tian J, Song L (2022) Driving distraction at night: The impact of cell phone use on driving behaviors among young drivers. Transport Res F: Traffic Psychol Behav 91:401–413. https://doi.org/10.1016/j.trf.2022.10.015
30. Ziryawulawo A, Kirabo M, Mwikirize C, Serugunda J, Mugume E, Miyingo SP (2023) Machine learning based driver monitoring system: A case study for the kayoola evs. SAIEE Africa Research Journal 114(2):40–48. https://doi.org/10.23919/SAIEE.2023.10071976
31. Zhang P, White J, Schmidt DC, Lenz G, Rosenbloom ST (2018) FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data. https://doi.org/10.48550/ARXIV.1807.03227

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Odinachi Udemezuo Nwankwo earned his undergraduate degree in Electrical and Electronic Engineering from the Federal University of Technology, Owerri, in 2014. He completed his Master's Degree in IT Convergence Engineering at Kumoh National Institute of Technology, South Korea in 2024. His research interests include networked embedded systems, IoT, AI, and blockchain.

Muhammad Rasyid Redha Ansori is currently working as a researcher at NSLab Inc. located in South Korea. He holds a Master's degree in IT Convergence Engineering from Kumoh National Institute of Technology, South Korea in 2023, and obtained his Bachelor's degree in Telecommunication Engineering from Telkom University, Indonesia in 2020. His research interests include blockchain, smart contracts, and real-time systems.

Gifar Arif Haryadi received the B.S. degree in telecommunication engineering from Telkom University, Bandung, Indonesia, in 2021, and the M.S. degree in IT convergence engineering from Kumoh National Institute of Technology, Gumi, South Korea, in 2025. He is currently pursuing the Ph.D. degree in IT convergence engineering at the same institution and is a researcher with the ICT Convergence Research Center. His research interests include blockchain technology, information security, and artificial intelligence.

Dong-Seong Kim (Senior Member, IEEE) received his Ph.D. degree in Electrical and Computer Engineering from Seoul National University, Seoul, Korea, in 2003. From 1994 to 2003, he worked as a full-time researcher at ERC-ACI at Seoul National University. From March 2003 to February 2005, he served as a postdoctoral researcher at the Wireless Network Laboratory in the School of Electrical and Computer Engineering at Cornell University, NY. Between 2007 and 2009, he was a visiting professor in the Department of Computer Science at the University of California, Davis, CA. He served as Dean of IACF from 2019 to 2022. He is currently a Professor with the Department of IT Convergence Engineering, School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, South Korea. He is also the director of the KIT Convergence Research Institute and the ICT Convergence Research Center (ITRC and NRF advanced research center program), supported by the Korean government at Kumoh National Institute of Technology, and the director of NSLab Inc. He is a senior member of IEEE and ACM. His primary research interests include realtime IoT and smart platforms, industrial wireless control networks, networked embedded systems, fieldbus, metaverse, and blockchain.

Jae Min Lee received his Ph.D degree in electrical and computer engineering from the Seoul National University, Seoul, Korea, in 2005. From 2005 to 2014, he was a Senior Engineer with Samsung Electronics, Suwon, Korea. From 2015 to 2016, he was a Principle Engineer in Samsung Electronics, Suwon, Korea. Since 2017, he has been an Associate professor with School of Electronic Engineering and Department of IT-Convergence Engineering, Kumoh National Institute of Technology, Gyeongbuk, Korea. Since 2024, he has been the Director of Smart Defense Logistics Innovation Convergence Research Center(ITRC Program). He is a member of IEEE. He is the Executive Director of the Korean Institute of Communications and Information Sciences (KICS). His current main research interests are smart IoT convergence application, industrial wireless control network, UAV, Metaverse and Blockchain.