International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 4, July – August 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Javipra AI: A Confidence-Weighted Multimodal Sensor Fusion Framework for Autonomous, Privacy-Preserving Emergency Detection and Response on Edge Devices

Jahnavi Somaraju[1], Chethan Kumar Reddy P[2]
[1]Assistant Professor, Department of Artificial Intelligence and Data Science
[2]Final Year UG Student, Department of Artificial Intelligence
Aditya College of Engineering, Madanapalle, Andhra Pradesh, India
jahnavisomaraju.research@gmail.com, Chethanreddy6616@gmail.com

Abstract:

Emergency response applications overwhelmingly rely on a victim's ability to manually trigger a distress signal, an assumption that fails precisely in the scenarios where intervention matters most: when a person is unconscious, physically restrained, under active assault, or otherwise unable to reach a device. Prior work has advanced human activity recognition, speech emotion recognition, audio event detection, and keyword spotting largely as isolated, single-modality problems, leaving open the question of how their heterogeneous, individually unreliable outputs can be combined into a single, actionable, and privacy-preserving judgment. This paper addresses that gap by presenting Javipra AI, an autonomous, privacy-preserving emergency detection framework whose central contribution is the Javipra Confidence Fusion Engine (JCFE), a confidence-weighted multimodal sensor fusion mechanism that departs from conventional weighted-average fusion by explicitly modeling inter-module agreement and temporal persistence as first-class evidentiary signals, rather than treating every module's confidence as independently and identically informative. Rather than depending on a single modality or a manually activated switch, the framework continuously and locally analyzes microphone audio, accelerometer and gyroscope streams, and GPS context through five independent artificial intelligence modules, voice keyword spotting, speech emotion recognition, audio event detection, human activity recognition, and GPS context analysis, each contributing a calibrated confidence score. JCFE combines these scores with contextual risk weighting into a unified emergency probability, which an adaptive decision engine compares against a context- and history-sensitive threshold explicitly designed to suppress false alarms without sacrificing sensitivity to genuine danger. On confirmation of an emergency, the system autonomously shares live location, notifies pre-designated trusted contacts, records encrypted emergency audio, and continues heightened monitoring, entirely without requiring deliberate user action. The complete inference pipeline is designed to execute on-device using TensorFlow Lite, so that raw audio, motion, and location data need never leave the device except as part of a minimized, authenticated emergency alert, giving the framework a privacy posture that is structural rather than policy-based, while enabling low-latency, offline-capable, battery-conscious operation on commodity smartphone hardware. This paper details the system architecture, the mathematical formulation of the confidence, risk, fusion, and decision functions, the algorithmic realization of JCFE, a proposed implementation stack, and a staged experimental methodology, including ablation studies and baseline comparisons, intended to validate the framework's detection accuracy and false-alarm behavior prior to field deployment. By reframing personal safety as an autonomous multimodal inference problem rather than a manually triggered event, Javipra AI is positioned to extend meaningful protection to precisely the population of emergencies, assault, abduction, medical incapacitation, falls, and restraint, that existing manual-activation systems structurally cannot address.

Keywords—*Edge AI; multimodal sensor fusion; emergency detection; privacy-preserving computing; human activity recognition; speech emotion recognition; audio event detection; TensorFlow Lite; autonomous emergency response; context-aware risk assessment.*

I. INTRODUCTION

The past decade has witnessed a marked shift in how individuals seek protection from physical danger, driven by the near-universal ownership of smartphones, the proliferation of low-cost inertial and acoustic sensors, and the maturation of on-device artificial intelligence. Personal safety technology has evolved from static panic buttons and pendant alarms toward software-based emergency applications that promise continuous protection through a device the user already carries. Government agencies, university campuses, ride-hailing companies, and individual users have all invested in some form of digital emergency response tool, reflecting a broader recognition that public safety infrastructure alone cannot guarantee timely intervention when an individual is threatened, injured, or isolated. As artificial intelligence models capable of running efficiently on mobile hardware have become practical, the opportunity to move beyond passive alarm tools toward systems that can reason about a user's situation in real time has become increasingly credible, and increasingly necessary.

Despite this progress, the overwhelming majority of commercial and academic emergency-response applications share a single structural assumption: that the user will consciously recognize a threat, retain sufficient physical and cognitive control to operate their device, and choose to press a button, shake the phone, or issue a voice command to summon help. This assumption of manual

activation is the central point of failure in existing systems. It requires the victim to perform a deliberate, well-coordinated action at precisely the moment when they are least capable of performing one. An emergency, by definition, is a moment in which normal capacity is disrupted, whether by physical incapacitation, psychological shock, restraint, or the direct interference of an assailant. A safety system that depends on the victim to initiate its own protection is, in a meaningful sense, unavailable exactly when it is needed most.

This limitation is not hypothetical; it manifests across a wide range of real-world scenarios that recur in crime reports, emergency-medicine literature, and personal-safety research. During a physical assault, an attacker frequently seizes or disables the victim's phone before any alert can be raised, and the victim's hands are rarely free to operate a touchscreen. In cases of kidnapping or forced abduction, the perpetrator typically controls the victim's environment and possessions, so that any visible attempt to activate an alarm risks immediate escalation of violence. Acute medical emergencies such as a stroke, cardiac arrest, or severe allergic reaction can impair motor coordination, speech, or consciousness within seconds, leaving the victim unable to unlock a device, let alone navigate an application. Falls, particularly among elderly individuals or those with mobility impairments, often result in temporary or prolonged unconsciousness, disorientation, or an inability to reach a phone that has fallen out of reach. Unconsciousness itself, whether from trauma, fainting, intoxication, or a medical event, removes the possibility of any voluntary action entirely. Physical restraint, whether through binding, confinement, or being pinned down, similarly removes the physical capacity to interact with a device. Even in situations that do not involve direct physical incapacitation, victims frequently experience a panic or freeze response, a well-documented physiological reaction to acute threat in which the body's stress system suppresses deliberate, goal-directed action in favor of an instinctive immobility response. In each of these situations, the presence of a manual SOS button provides no practical benefit, because the precondition for using it, a calm, capable, and unencumbered user, is exactly what the emergency has taken away.

These failure modes motivate a fundamentally different design philosophy for personal safety systems: rather than waiting for the user to declare that an emergency has occurred, the system itself should be capable of recognizing that an emergency is likely occurring, based on observable evidence in the user's environment and physiology. This is the premise of autonomous emergency detection. Instead of a single explicit trigger, an autonomous system continuously and passively observes multiple channels of evidence, such as vocal distress, characteristic sounds of violence or falling, abnormal body motion, and unusual movement or location patterns, and reasons over this evidence to infer whether the user is likely in danger. Because no single sensor or signal is a reliable indicator of an emergency in isolation, autonomous detection is only practical when several independent sources of evidence can be combined in a manner that is robust to noise, ambiguity, and the ordinary variability of daily life.

Realizing this vision responsibly requires several complementary design principles. Edge AI, the execution of inference directly on the user's device rather than in a remote data center, is essential both for latency and for reliability: an emergency-detection system that depends on a network connection is unavailable in basements, elevators, moving vehicles, rural areas, or any location with unreliable connectivity, precisely the settings in which many emergencies occur. Privacy-preserving computing is equally essential, since a system that continuously listens to a user's surroundings and tracks their movement carries an inherent risk of becoming a surveillance liability if raw audio, sensor, or location data is transmitted or stored off-device by default; a responsible design must ensure that continuous monitoring does not become continuous exposure. Multimodal sensing is necessary because any individual modality, taken alone, produces an unacceptable rate of false positives and false negatives: a loud noise alone does not indicate assault, elevated heart rate alone does not indicate danger, and rapid motion alone does not indicate a fall. Sensor fusion, the principled combination of evidence across modalities, is what allows a system to distinguish a true emergency from the ordinary acoustic and physical noise of daily life. Finally, context-aware intelligence, the ability to weigh evidence differently depending on the user's situation, such as distinguishing vigorous exercise from a physical struggle, or a loud television from a genuine cry for help, is what allows an autonomous system to remain sensitive to real danger without becoming a source of constant false alarms.

It is within this design space that the present work situates Javipra AI. Javipra AI is proposed not as an incremental improvement to manually activated SOS applications, but as a distinct category of personal safety system: one in which emergency detection is an autonomous inference problem rather than a user-initiated event. At a conceptual level, Javipra AI continuously interprets evidence drawn from a user's voice, ambient sound, physical motion, and location context, and reasons over this evidence using a set of independent artificial intelligence modules, each of which contributes an assessment of how consistent the current situation is with an emergency. The manner in which these independent assessments are combined into a single, actionable judgment, and the mechanism by which the system decides when that judgment is confident enough to warrant an autonomous response, form the central technical contribution of this paper and are developed in detail in the sections that follow. The present section introduces the framework at a conceptual level; the architectural, mathematical, and algorithmic specifics are deferred to later sections so that the underlying design philosophy can first be established on its own terms.

The motivation for this research, therefore, is not the absence of emergency-response technology, but the structural inadequacy of the dominant paradigm within that technology. Manual-activation systems have undoubtedly saved lives in situations where a user retained the ability to act, but their protective value collapses precisely in the categories of emergency, assault, abduction, medical incapacitation, restraint, and panic, that are often the most severe and the most time-critical. Closing this gap requires safety systems capable of recognizing danger independently of the victim's ability to ask for help, while doing so in a way that respects user privacy, operates reliably without network dependence, and avoids the false-alarm fatigue that would render a hypersensitive system unusable in practice.

The research problem addressed in this paper can accordingly be stated as follows: how can heterogeneous, noisy, and individually ambiguous streams of sensor evidence, namely audio, motion, and location, be combined on a resource-constrained edge device to infer, with sufficient confidence and sufficiently low false-alarm rate, that a user is experiencing a genuine emergency, and how can this inference be translated into an autonomous protective response without requiring any deliberate action from the user and without compromising the privacy of their continuously monitored data. This problem sits at the intersection of multimodal machine learning, edge computing, and human-centered safety system design, and its solution requires contributions at each of these levels rather than at any one of them in isolation.

In addressing this problem, this paper makes the following principal contributions:

- **A novel Confidence-Weighted Multimodal Sensor Fusion Framework:** a unified architecture that integrates voice, ambient audio, motion, and location evidence through independently generated, per-module confidence scores rather than a single monolithic classifier.
- **The Javipra Confidence Fusion Engine (JCFE):** a new fusion algorithm that combines heterogeneous, independently produced AI outputs into a single, calibrated emergency probability, forming the central novelty of this work.
- **Privacy-preserving, on-device Edge AI inference:** a design in which all sensor interpretation executes locally using lightweight, quantized models, so that raw audio, motion, and location data need not leave the user's device under normal operation.
- **Adaptive emergency decision-making for false-alarm reduction:** a decision layer that adjusts sensitivity to situational and contextual factors, reducing the rate of spurious alerts that would otherwise undermine user trust and system adoption.
- **Autonomous emergency response without manual user intervention:** an end-to-end response pipeline, encompassing trusted-contact notification, live location sharing, and encrypted evidence capture, that is triggered entirely by the system's own inference rather than by any explicit user action.

The remainder of this paper is organized as follows. Section II reviews related work in emergency-detection systems, multimodal sensor fusion, and edge-deployed artificial intelligence, and identifies the specific research gaps that motivate this work. Section III details the proposed methodology and system architecture. Section IV presents the mathematical formulation underlying the confidence, risk, and fusion computations. Section V describes the algorithmic realization of each module. Subsequent sections address implementation, experimental methodology, discussion, limitations, and conclusions.

II. LITERATURE REVIEW

The research problem motivating this work sits at the confluence of several previously distinct fields: manually activated personal safety applications, single-modality AI-based hazard detectors, multimodal sensor fusion, and edge-deployed machine learning. This section reviews each of these strands in turn, before Section III consolidates the specific gaps that remain when they are considered together.

A. Manually Activated Emergency and Personal Safety Applications

The dominant paradigm in commercial personal-safety software is the manually triggered alert. WoSApp allows a user to discreetly place an emergency call to the police by shaking the phone or pressing an on-screen panic button, transmitting the user's location and a pre-selected list of emergency contacts [25]. SAKHI similarly requires the user to hold a volume button for a sustained duration to broadcast an SOS alert to trusted contacts and the police [25]. Comparable systems reviewed in the women's-safety literature, including shake-activated and single-tap alert applications, uniformly depend on the user recognizing danger and performing a deliberate physical action to raise an alarm. Systematic reviews of this class of application consistently identify the same shortcoming: because activation is manual, systems fail precisely when the user is incapacitated, restrained, or otherwise unable to interact with the device, and false alarms or delayed activation remain persistent practical problems. These findings directly motivate the case for autonomous, sensor-driven detection rather than incremental improvement of manual-trigger mechanisms.

B. Human Activity Recognition and Fall Detection

Human activity recognition (HAR) using smartphone and wearable inertial sensors has matured into an extensive research area. Gu et al. provide a comprehensive survey of deep learning approaches to HAR, cataloguing convolutional, recurrent, and hybrid architectures applied to accelerometer and gyroscope streams [1]. Demrozi et al. similarly survey HAR using inertial, physiological, and environmental sensors, highlighting the trade-offs between recognition accuracy, sensor placement, and computational cost [2]. Ramanujam et al. review deep learning techniques specifically for smartphone- and wearable-based HAR, noting the shift from handcrafted statistical features toward end-to-end learned representations [3]. Ronao and Cho demonstrated that convolutional neural networks applied directly to raw smartphone accelerometer and gyroscope signals outperform classifiers built on manually engineered features [4], while Xia et al. proposed a combined LSTM-CNN architecture that captures both spatial and temporal patterns in inertial data with improved recognition accuracy [5]. Within this broader HAR literature, fall detection has received particular attention because of its direct relevance to personal safety: Rakhman et al. implemented a smartphone-based fall detector using accelerometer and gyroscope thresholding to trigger an automatic alert call [20], and De Cillis et al. proposed a fall-detection algorithm for mobile platforms evaluated on both accelerometer and gyroscope features, reporting the practical challenges of distinguishing genuine falls from similarly abrupt but benign movements such as sitting down quickly or placing a phone on a table [21]. Dian et al. further situate these sensor-based approaches within the broader landscape of wearable and Internet-of-Things devices, surveying the opportunities and constraints of continuous on-body sensing [6].

C. Speech and Audio-Based Emergency Cues

A parallel body of work addresses the recognition of distress directly from acoustic signals. Speech emotion recognition (SER) research has progressively moved from handcrafted prosodic and spectral features toward deep representation learning: Issa et al. applied one-dimensional convolutional neural networks directly to raw speech features to classify emotional state with improved accuracy over classical pipelines [7], Er proposed a hybrid deep and acoustic feature approach for emotion classification [8], and Mukhamediya et al. examined how log-mel spectrogram parameterization affects the performance of deep SER models, underscoring the sensitivity of such systems to front-end signal processing choices [9]. Beyond speech content and emotion, a distinct line of research addresses the detection of non-verbal acoustic events indicative of danger: Valenzise et al. developed an audio-surveillance system that detects and localizes screams and gunshots in public spaces using parallel Gaussian mixture model classifiers, reporting detection precision above 90% under moderate signal-to-noise conditions [10]. This scream- and gunshot-detection literature demonstrates that non-speech acoustic events carry strong, learnable evidence of physical danger, complementary to the semantic and emotional content of speech. Keyword spotting research, meanwhile, has focused on recognizing specific spoken triggers with minimal computational footprint: Chen et al. introduced a small-footprint deep neural network for keyword spotting suitable for always-on listening [11], Sainath and Parada proposed a convolutional variant with improved accuracy at comparable model size [13], Zhang et al. demonstrated that such keyword spotters can be compressed to run continuously on microcontroller-class hardware [12], and López-Espejo et al. provide a comprehensive overview of the deep spoken keyword spotting literature, tracing the field's progress toward models suitable for always-on, battery-constrained deployment [14].

D. Multimodal Sensor Fusion

Because no individual sensing modality reliably distinguishes a genuine emergency from ordinary daily activity, an extensive body of research addresses the combination of heterogeneous signals into a unified inference. Baltrušaitis et al. provide the foundational taxonomy of multimodal machine learning, formalizing the distinction between early, late, and hybrid fusion strategies and cataloguing the technical challenges of representation, alignment, and co-learning across modalities [22]. This taxonomy is directly applicable to the problem addressed in this paper, since audio, inertial, and location evidence differ substantially in sampling rate, noise characteristics, and semantic granularity, precisely the kind of heterogeneity that motivates principled fusion architectures rather than naive concatenation of features. The broader HAR literature has increasingly adopted multimodal approaches for the same reason: combining multiple wearable and environmental sensors has been shown to substantially improve robustness over single-sensor recognition, particularly in ambiguous or noisy real-world conditions.

E. Edge AI and TinyML for On-Device Inference

The practicality of continuous, always-on AI inference on personal devices depends on the maturation of edge and tiny machine learning. Chen and Ran review the general landscape of deep learning with edge computing, framing the trade-offs between model accuracy, latency, and on-device resource constraints [18]. Sze et al. provide a widely cited tutorial on efficient neural network processing, detailing the hardware and algorithmic techniques, including quantization and pruning, that make deep inference tractable on constrained devices [19]. More recent surveys focus specifically on TinyML: Abadade et al. survey the deployment of machine learning models on microcontroller-class hardware operating under severe memory and power budgets [15], Capogrosso et al. similarly catalogue the software toolchains and hardware platforms that support tiny machine learning deployment [16], and Saha et al. review machine learning specifically for microcontroller-class hardware, highlighting the specific optimizations required for continuous sensing applications such as keyword spotting and activity recognition [17]. Collectively, this literature establishes that the computational techniques required for continuous, on-device, multimodal inference, quantized convolutional and recurrent architectures executed within kilobytes to low megabytes of memory, are now practical on commodity smartphone hardware.

F. Privacy-Preserving Mobile Sensing

Because continuous audio and motion sensing carries inherent privacy risk, a further relevant literature addresses privacy-preserving computation for mobile and edge data. Kairouz et al. provide an extensive treatment of federated learning, cataloguing techniques for training and adapting models across distributed devices without centralizing raw user data [23], and Li et al. review the practical challenges of federated learning, including statistical heterogeneity, communication efficiency, and privacy guarantees, in a form directly relevant to any system that seeks to improve its models without transmitting raw sensor recordings to a server [24]. Together with the edge AI literature reviewed above, this work establishes both the motivation and the technical foundation for a system in which sensor interpretation occurs locally and only high-level, privacy-preserving outputs, rather than raw audio or location traces, are ever considered for transmission.

III. RESEARCH GAP ANALYSIS

Considered individually, the six strands of literature reviewed in Section II are each substantially developed. Considered together, however, they reveal a set of gaps that no existing system fully addresses, and which collectively define the contribution of this paper.

TABLE 1
Summary of Identified Research Gaps

Research Area	State of the Art	Identified Gap
Emergency-response applications	Manual SOS activation via button press, shake, or voice command [25]	No mechanism for autonomous detection when the user cannot act
Human activity recognition / fall detection	High-accuracy recognition of falls and activities from inertial sensors [1]–[5], [20], [21]	Evaluated as a standalone signal; not integrated into a broader emergency-inference pipeline
Speech and audio event detection	Effective isolated detection of emotional speech, keywords, or acoustic events [7]–[14]	Each modality developed and validated independently, without a mechanism for joint reasoning
Multimodal sensor fusion	General-purpose fusion taxonomies and architectures [22]	Not previously specialized for the emergency-detection domain or for confidence-weighted, module-level combination
Edge AI / TinyML	Demonstrated feasibility of on-device inference for individual tasks [15]–[19]	Not applied to a coordinated, multi-module emergency-inference pipeline running concurrently on a single device
Privacy-preserving computation	Federated and on-device learning frameworks [23], [24]	Not incorporated into the design of continuous personal-safety monitoring systems

First, the manual-activation gap: every reviewed personal-safety application, irrespective of its specific trigger mechanism, requires a deliberate user action, and consequently inherits the structural failure mode discussed in Section I. Second, the modality-isolation gap: HAR, fall detection, speech emotion recognition, keyword spotting, and acoustic event detection have each been developed and validated as independent research problems, typically evaluated on modality-specific benchmark datasets, with little attention paid to how their outputs might be combined into a single actionable judgment about a shared, real-world situation. Third, the domain-specialization gap in fusion research: while general multimodal fusion taxonomies are well established [22], no framework in the reviewed literature specializes confidence-weighted fusion for the specific evidentiary structure of emergency detection, in which modalities differ sharply in reliability depending on context, and in which the cost of a false negative (a missed emergency) is far higher than the cost of a false positive, an asymmetry that generic fusion objectives do not explicitly encode. Fourth, the integration gap in edge deployment: although individual modules such as keyword spotting or activity recognition have been demonstrated on constrained hardware [11], [12], [15]–[17], no reviewed work integrates several such modules into a single, concurrently executing, on-device pipeline feeding a shared decision engine. Fifth, the privacy-by-design gap: privacy-preserving computation techniques exist as a general research area [23], [24] but have not been incorporated into the architectural design of autonomous personal-safety systems, which as currently constructed offer no structural guarantee that continuously sensed audio, motion, or location data remains on-device.

Javipra AI is proposed specifically to close these five gaps jointly, rather than to advance any one of the underlying component technologies in isolation. Its contribution is therefore architectural and integrative as much as algorithmic: the Javipra Confidence Fusion Engine (JCFE), introduced formally in Section VII, is designed as a domain-specific fusion mechanism that treats false-negative avoidance and per-module reliability as first-class design considerations, operating entirely on-device over the outputs of concurrently executing HAR, speech, and audio-event modules, without requiring raw sensor data to leave the device under normal operation.

IV. PROPOSED METHODOLOGY

The methodology adopted in this work follows directly from the research gaps identified in Section III. Rather than proposing a new single-modality classifier, the methodology centers on the design of an integrative, on-device architecture in which independently trained perception modules feed a shared, confidence-weighted fusion and decision layer. Three methodological principles guide this design: modularity, so that each perception module can be developed, trained, and improved independently using the modality-specific techniques established in the literature reviewed in Section II; confidence-based abstraction, so that the fusion layer reasons over calibrated confidence scores rather than raw sensor data, keeping the fusion mechanism agnostic to the internal implementation of any individual module; and privacy-by-architecture, so that every stage of the pipeline, from raw signal acquisition through fusion and decision-making, executes locally on the user's device, with network transmission reserved exclusively for the final, high-level output of an emergency alert.

The overall methodology proceeds in five stages. The first stage, sensing, continuously acquires microphone audio, tri-axial accelerometer and gyroscope readings, and coarse-grained GPS context, using duty-cycled sampling to conserve battery. The second stage, perception, applies a set of independent, lightweight AI modules, voice keyword spotting, speech emotion recognition, audio event detection, human activity recognition, and GPS context analysis, each of which produces both a classification output and an associated confidence score reflecting the module's own certainty. The third stage, contextual risk assessment, adjusts the interpretation of these per-module outputs according to situational context, such as time of day, location category, and recent activity history, in order to distinguish, for example, elevated heart rate and rapid motion during exercise from the same signature occurring

during a physical struggle. The fourth stage, confidence-weighted fusion, is performed by the Javipra Confidence Fusion Engine, which combines the per-module confidence scores and contextual risk adjustment into a single, calibrated emergency probability. The fifth stage, adaptive decision-making, compares this probability against a context-sensitive threshold to determine whether the evidence is sufficient to trigger an autonomous emergency response, and if not, whether it warrants continued heightened monitoring.

This staged methodology is deliberately structured so that later sections can treat each stage with the appropriate level of rigor: Section V describes the architectural realization of these five stages as system components; Section VI describes the operational workflow that governs their real-time interaction; Section VII formalizes the mathematical relationships between confidence, risk, fusion, and decision; and Section VIII provides the algorithmic pseudocode that implements this mathematical model.

V. SYSTEM ARCHITECTURE

The Javipra AI system architecture is organized into five layers, corresponding to the five methodological stages introduced in Section IV: a Sensing Layer, a Perception Layer, a Context and Risk Layer, a Fusion Layer, and a Response Layer. Fig. 1 depicts the overall architecture and the flow of information between these layers.

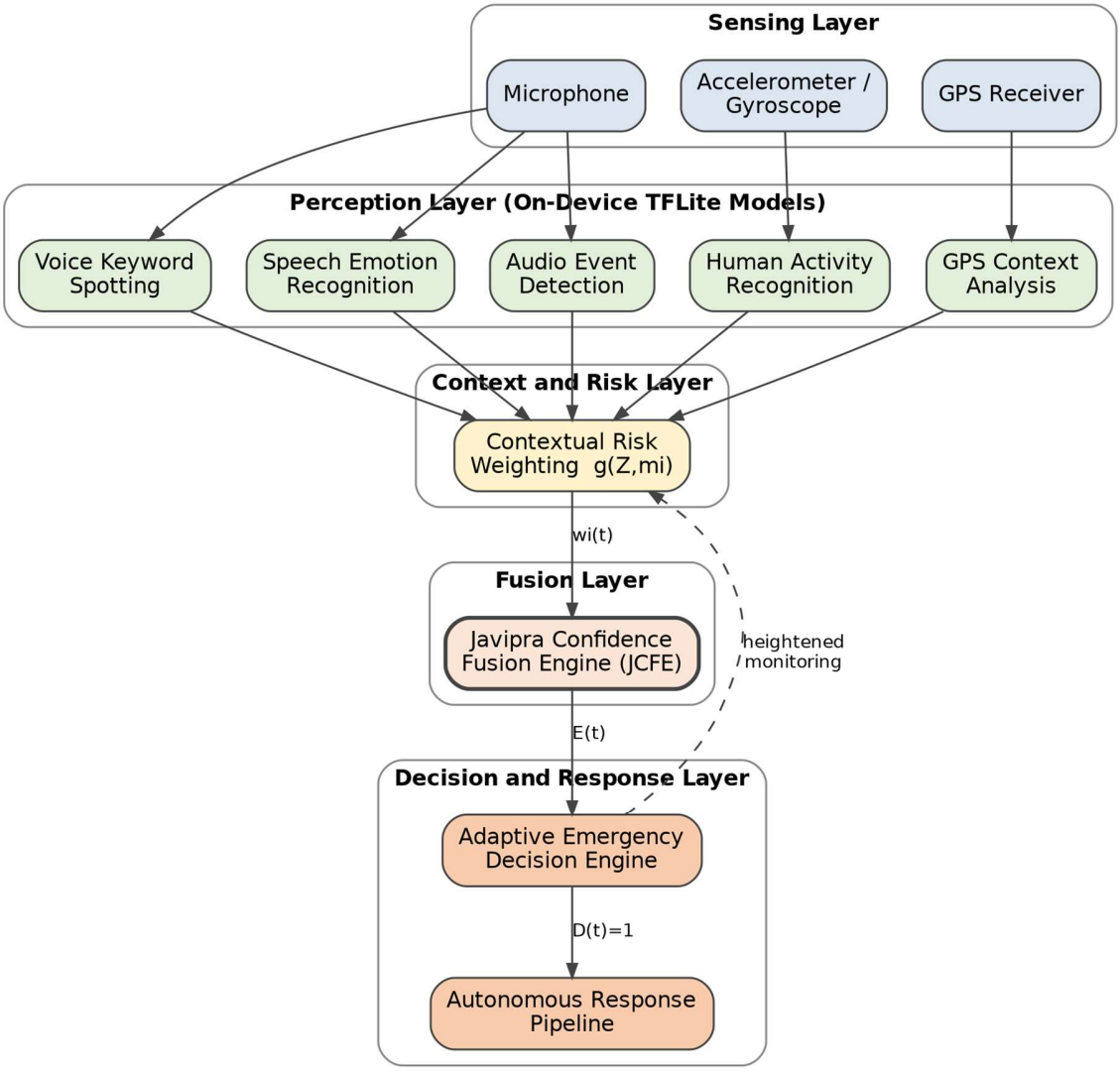

Fig. 1. Overall system architecture of Javipra AI, showing the Sensing, Perception, Context and Risk, Fusion, and Decision/Response layers and the flow of confidence scores between them.

A. Sensing Layer

The Sensing Layer is responsible for acquiring raw signals from the device's built-in microphone, tri-axial accelerometer, tri-axial gyroscope, and GPS receiver. To conserve battery, sensors are duty-cycled: audio is analyzed in short overlapping windows, inertial sensors are sampled continuously at a moderate rate sufficient for activity and fall recognition, and GPS is polled at a lower frequency, with update rate increased

only once the Fusion Layer registers elevated risk. Raw data at this layer never leaves the device and is discarded once the corresponding Perception Layer module has produced its output.

B. Perception Layer

The Perception Layer hosts five independent AI modules, each realized as a lightweight, quantized on-device model. The Voice Keyword Spotting module continuously screens short audio frames for a small set of distress-indicative keywords and phrases, following the small-footprint deep neural network design established by Chen et al. and later adapted for microcontroller-class deployment [11], [12]. The Speech Emotion Recognition module analyzes prosodic and spectral features of detected speech segments to estimate the likelihood of fear, distress, or panic, drawing on convolutional architectures shown effective for on-device emotion classification [7]–[9]. The Audio Event Detection module screens ambient sound for acoustic signatures associated with physical danger, such as screaming, shouting, or the sound of a struggle, following the detection methodology established for scream and impulsive-event recognition [10]. The Human Activity Recognition module classifies the user's current physical activity and detects fall-like or restraint-like motion signatures from accelerometer and gyroscope streams, building on convolutional and hybrid CNN-LSTM architectures shown effective for smartphone-based HAR [1]–[5], [20], [21]. The GPS Context Analysis module evaluates location-derived features, such as deviation from typical movement patterns, unusually prolonged stationarity in a non-residential location, or unexpected excursion outside a habitual radius. Each module outputs both a class label and a per-module confidence score in the interval [0, 1], reflecting its own certainty in that output; the internal computation of this score is module-specific and is treated as opaque by the layers above.

C. Context and Risk Layer

The Context and Risk Layer receives the outputs of the Perception Layer together with situational metadata, including time of day, location category (e.g., residential, transit, unfamiliar), recent activity history, and user-configured context such as scheduled exercise periods. This layer computes a contextual risk multiplier that adjusts the weight given to each module's confidence score before fusion, allowing the system to, for example, discount elevated heart-rate-correlated motion signatures during a known exercise window, while giving full weight to the same signature occurring at an atypical hour in an atypical location.

D. Fusion Layer: The Javipra Confidence Fusion Engine

The Fusion Layer implements the Javipra Confidence Fusion Engine (JCFE), the central technical contribution of this paper. JCFE accepts the confidence-weighted, context-adjusted outputs of all active Perception Layer modules and combines them into a single, calibrated Emergency Probability, formalized in Section VII. Because the number and combination of active modules can vary from moment to moment, for example, when no speech is present and only motion and location evidence are available, JCFE is designed to operate over a variable-length set of module outputs rather than a fixed-dimensional feature vector.

E. Response Layer

When the Emergency Probability produced by JCFE exceeds the adaptive decision threshold described in Section VII, the Response Layer autonomously executes the emergency response pipeline detailed in Section VI-D: sharing the user's live GPS location, notifying pre-designated trusted contacts, beginning encrypted local recording of ambient audio as evidentiary documentation, and continuing heightened monitoring until the situation resolves or is manually cleared by the user. Notification payloads are limited to the minimum information necessary for effective response, location, timestamp, and a brief system-generated summary, rather than raw sensor recordings.

F. Component View

Fig. 2 decomposes the architecture into concrete software components. The Flutter application shell delegates sensor access to a native Sensor Manager, which feeds the TensorFlow Lite inference engine hosting the five quantized Perception Layer models. Module outputs converge on the Context Engine, then the JCFE module, then the Decision Engine, and finally the Response Manager, which is the only component authorized to write to encrypted local storage or to invoke Firebase Cloud Messaging. This separation of concerns confines all raw-data-handling components to the left-hand side of Fig. 2, while only alert-level, minimized payloads cross the component boundary into cloud-facing services on the right.

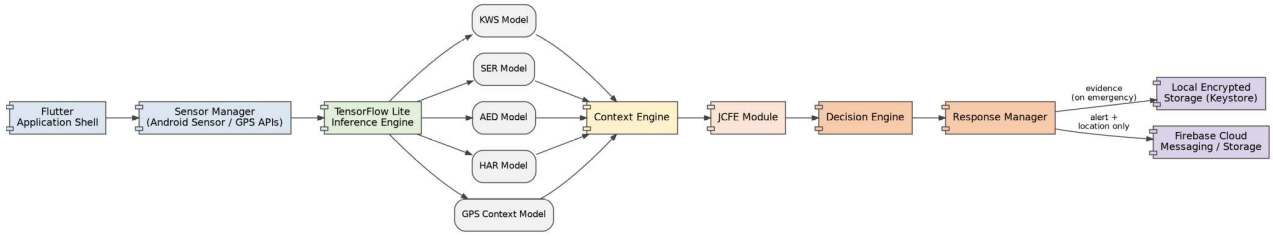

Fig. 2. Component diagram of Javipra AI, decomposing the architecture of Fig. 1 into concrete software components and their interfaces.

G. Deployment View

Fig. 3 shows the physical deployment of Javipra AI. The complete perception, fusion, and decision pipeline is deployed entirely on the user's smartphone; the only cloud-facing component is a minimal, alert-only messaging service used exclusively to relay the final emergency notification and, optionally, an encrypted evidence backup initiated by the user rather than by the system autonomously. No component in the deployment topology requires continuous transmission of raw audio, motion, or location data, so the system remains fully functional for detection purposes even when the device is offline; only the final notification step requires connectivity.

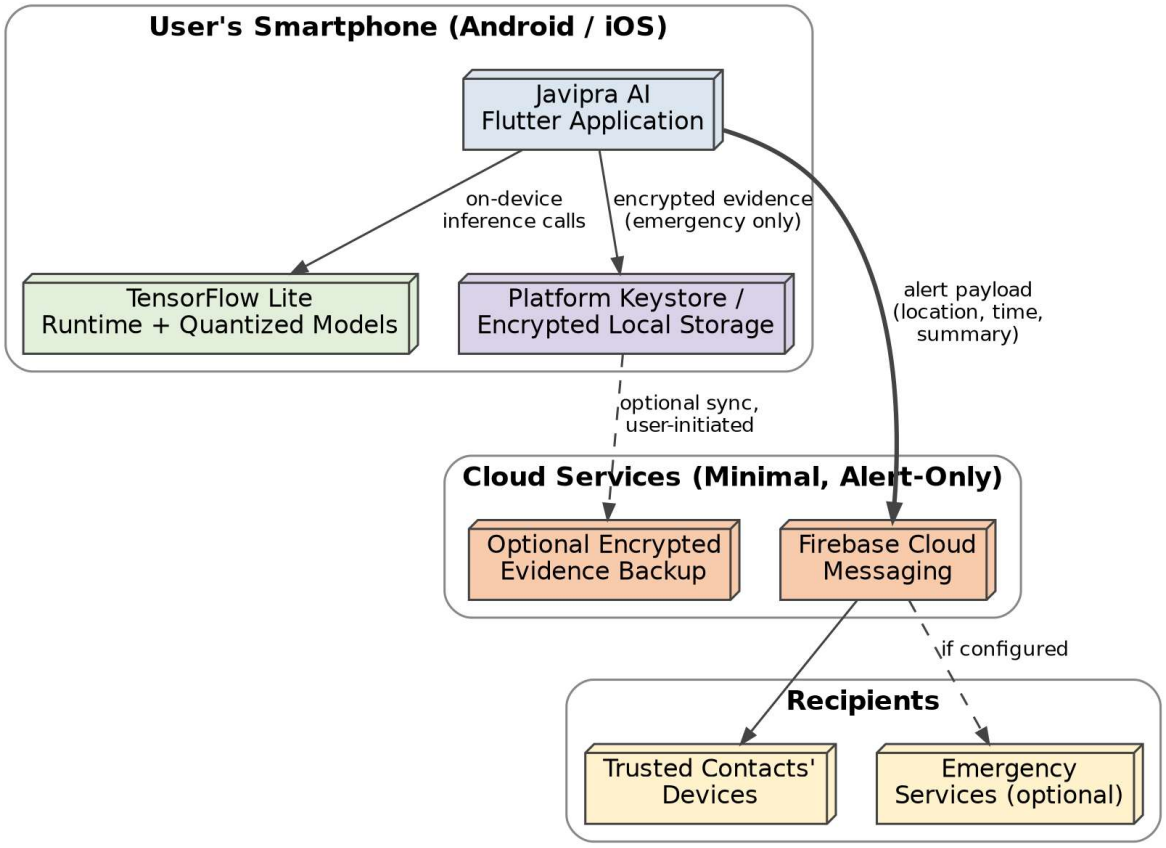

Fig. 3. Deployment diagram of Javipra AI, showing the on-device inference boundary and the minimal, alert-only cloud interaction with trusted contacts and emergency services.

VI. WORKFLOW

This section operationalizes the architecture of Section V as a continuous runtime process, and presents four complementary views: the control-flow workflow (Fig. 4), the data-flow and privacy boundary (Fig. 5), the inter-component interaction sequence (Fig. 6), and the emergency response pipeline invoked once a decision is confirmed (Fig. 7).

A. Continuous Monitoring Workflow

Fig. 4 illustrates the end-to-end operational workflow of Javipra AI, corresponding to the continuous background process executed on the user's device.

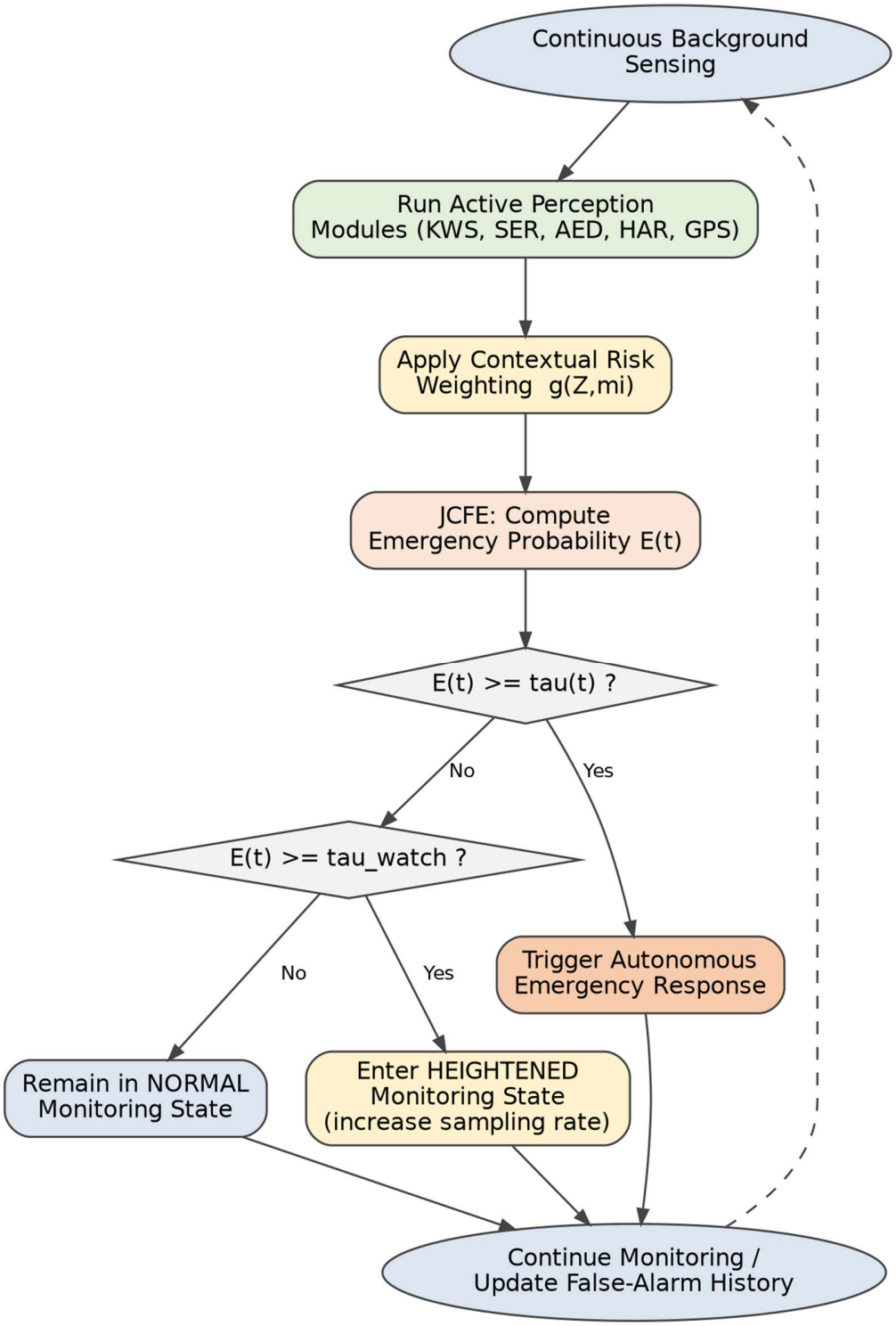

Fig. 4. End-to-end operational workflow of Javipra AI, from continuous sensor acquisition through perception, fusion, adaptive decision, and autonomous response.

The workflow begins with continuous background acquisition of microphone audio, accelerometer data, gyroscope data, and GPS information by the Sensing Layer. Each acquired signal is routed to its corresponding Perception Layer module: audio frames to the Voice Keyword Spotting, Speech Emotion Recognition, and Audio Event Detection modules; inertial frames to the Human Activity Recognition module; and location updates to the GPS Context Analysis module. Each module produces a class output and confidence score, which the Context and Risk Layer adjusts according to the current situational context. The adjusted, per-module confidence scores are passed to the Javipra Confidence Fusion Engine, which computes a unified Emergency

Probability E(t). The Adaptive Emergency Decision Engine compares this probability against the current adaptive threshold τ(t). If E(t) remains below the secondary watch threshold, the system returns to the NORMAL monitoring state. If E(t) exceeds the watch threshold but not τ(t), the system enters a HEIGHTENED_MONITORING state, retaining a short rolling buffer of recent module outputs and increasing sampling rate so that evidence accumulating gradually over several seconds, rather than arising from a single instantaneous spike, can still be recognized. If E(t) meets or exceeds τ(t), the workflow proceeds to the Emergency Response Pipeline detailed in Section VI-D.

B. Data Flow and Privacy Boundary

Fig. 5 makes explicit the trust boundary that underlies the privacy-preserving design principle of Section IV. Raw audio, inertial, and GPS buffers are consumed by the Perception Layer modules and discarded immediately after inference; only the resulting confidence scores, which are compact numerical summaries rather than reconstructable recordings, cross into the Fusion and Decision layers. Data leaves the device in only two circumstances, both gated by D(t) = 1: an encrypted evidence recording written to local storage, and a minimized alert payload, containing location, timestamp, and a system-generated summary rather than raw audio, transmitted to trusted contacts. This diagram is intended to make the system's privacy posture independently auditable: any data path that crosses the dashed trust boundary in Fig. 5 without passing through the D(t) = 1 gate would represent a deviation from the intended design.

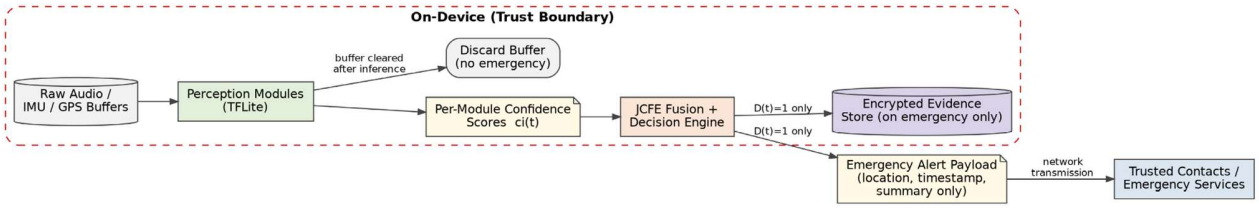

Fig. 5. Data flow diagram of Javipra AI, showing the on-device trust boundary and the two gated conditions under which any information leaves the device.

C. Interaction Sequence

Fig. 6 presents the interaction sequence among architectural components for a single detection cycle that culminates in a confirmed emergency. The Sensing Layer forwards a synchronized frame to the Perception Modules (message 1), which return a class label and confidence score to the Context and Risk Layer (message 2). The Context and Risk Layer computes the contextual weight and forwards it to JCFE (message 3), which returns the fused Emergency Probability to the Decision Engine (message 4). The Decision Engine performs the threshold comparison internally (message 5) and, upon confirming an emergency, signals the Response Manager (message 6), which notifies the trusted contact (message 7) and begins encrypted evidence capture (message 8). When E(t) does not cross τ(t), the sequence instead terminates after message 5 and the cycle repeats at the next sensing interval.

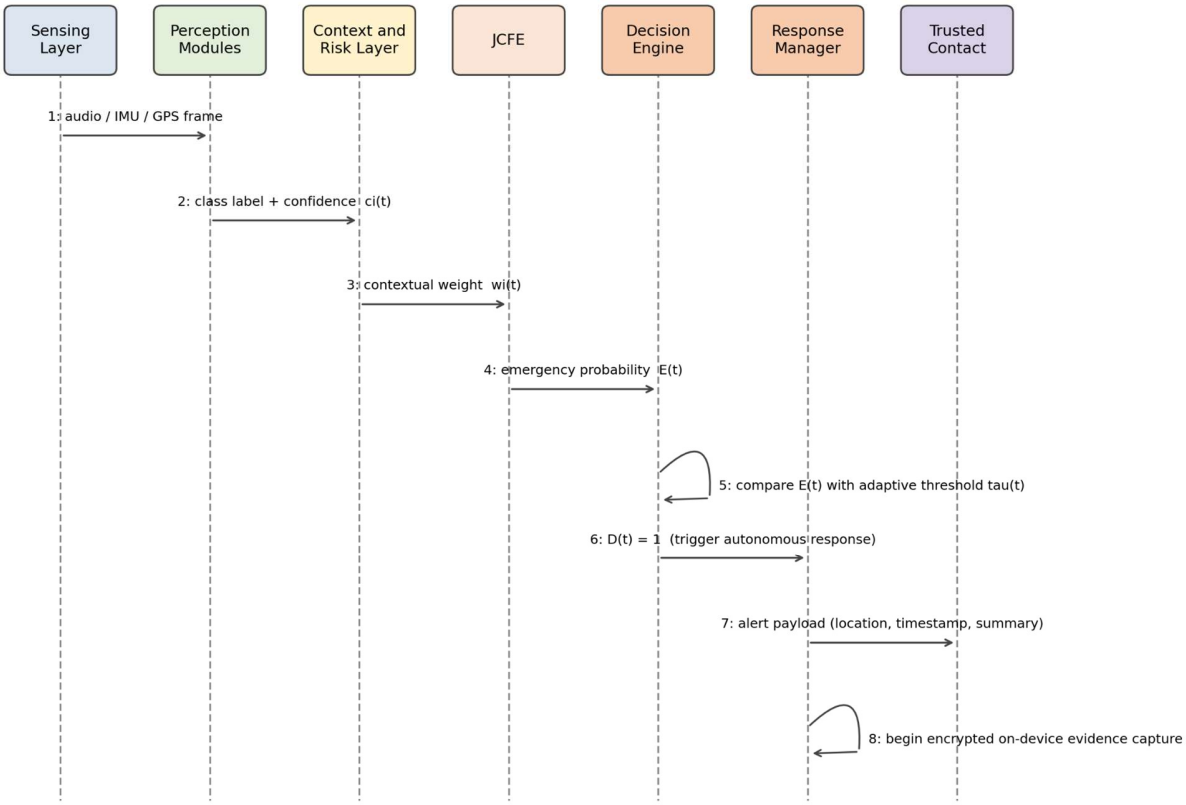

Fig. 6. Sequence diagram of a single Javipra AI detection cycle culminating in a confirmed emergency response.

D. Emergency Response Pipeline

Fig. 7 details the response pipeline invoked once the Adaptive Emergency Decision Engine sets D(t) = 1. Rather than executing response actions sequentially, the pipeline forks into three actions performed concurrently, live location sharing, trusted-contact notification, and encrypted evidence recording, so that the overall response latency is bounded by the slowest individual action rather than their sum. Once all three actions have been initiated, the pipeline joins into a heightened-monitoring state in which captured evidence continues to be stored on-device and the system continues monitoring until the emergency is resolved or manually cleared by the user, at which point the system returns to NORMAL monitoring.

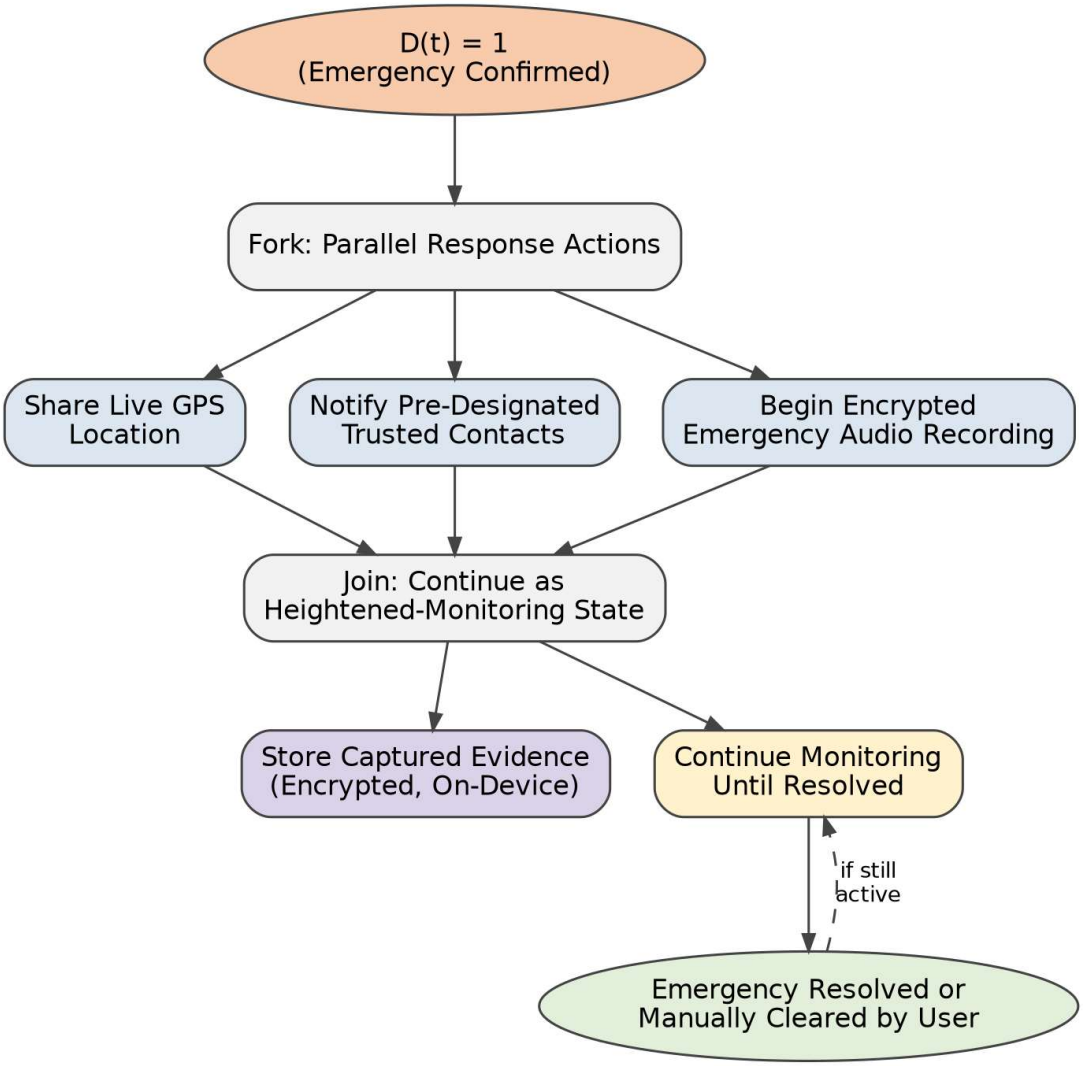

Fig. 7. Emergency response pipeline of Javipra AI, showing the concurrent fork of response actions triggered by D(t) = 1 and the subsequent join into heightened monitoring.

VII. MATHEMATICAL MODEL

This section formalizes the confidence, risk, fusion, and decision computations introduced conceptually in Sections IV through VI. Let M = {m1, m2, ..., mk} denote the set of Perception Layer modules that are active at a given time instant t (k varies because not every module produces output at every instant, for example, when no speech is present).

A. Per-Module Confidence Score

Each active module mi produces a confidence score ci(t) ∈ [0, 1] at time t, representing that module's own estimate of the likelihood that its observed evidence is consistent with an emergency. For a module implemented as a neural classifier, ci(t) is typically derived from the softmax output associated with the emergency-consistent class, though the internal computation is module-specific and is treated as opaque by the Fusion Layer.

$$ci(t) = P(\text{emergency-consistent evidence} \mid xi(t)),\ ci(t) \in [0, 1] \quad (1)$$

B. Contextual Risk Weight

The Context and Risk Layer assigns each module a contextual weight wi(t) ∈ [0, 1] that reflects the reliability of that module's evidence given the current situational context Z(t), such as time of day, location category, and recent activity history. The weight is computed as:

$$wi(t) = ri \cdot g(Z(t), mi) \quad (2)$$

where ri is a fixed base reliability coefficient for module mi, reflecting its typical precision as established during offline validation, and g(Z(t), mi) ∈ [0, 1] is a context modulation function that discounts a module's contribution when the current context provides an alternative, benign explanation for its evidence (for example, discounting elevated motion-derived confidence during a user-declared exercise window).

C. Risk Score

The overall risk score R(t) aggregates the contextually weighted confidence scores of all active modules prior to fusion:

$$R(t) = \Sigma i=1..k\ [wi(t) \cdot ci(t)] \ / \ \Sigma i=1..k\ wi(t) \quad (3)$$

R(t) thus represents a weighted average of module confidences, normalized by total active weight so that the risk score remains on the [0, 1] scale regardless of how many modules are currently active.

D. Javipra Confidence Fusion Engine (JCFE)

The central fusion computation, performed by JCFE, extends the weighted average of Eq. (3) with two additional terms that reflect the specific evidentiary structure of emergency detection: an agreement bonus that increases the fused probability when multiple independent modules agree, since independent corroboration is stronger evidence than any single high-confidence module, and a persistence term that reflects whether elevated confidence has been sustained or growing over a short recent time window, rather than arising from a single transient spike. The Emergency Probability E(t) is defined as:

$$E(t) = \sigma(\ \alpha \cdot R(t) + \beta \cdot A(t) + \gamma \cdot P(t) - \delta\ ) \quad (4)$$

where σ(·) denotes the logistic function mapping its argument to the [0, 1] interval; R(t) is the risk score from Eq. (3); A(t) ∈ [0, 1] is an inter-module agreement term, computed as one minus the normalized variance of the active ci(t) values, so that A(t) is high when active modules concur and low when they conflict; P(t) ∈ [0, 1] is a persistence term computed as a short exponentially weighted moving average of R(t) over the preceding time window, rewarding sustained or escalating evidence over isolated spikes; α, β, and γ are non-negative weighting coefficients satisfying α + β + γ = 1, determined during offline validation to balance sensitivity against false-alarm rate; and δ is a calibration offset that centers the logistic function at the desired baseline operating point.

JCFE is deliberately distinguished from three conventional alternatives. First, a simple weighted average, E(t) = R(t), of the kind common in early sensor-fusion literature, treats every instant of evidence as equally trustworthy regardless of whether it is corroborated or transient; JCFE's agreement term A(t) instead rewards corroboration and its persistence term P(t) discounts single-instant spikes, which is essential in this domain because a single mistaken module output (e.g., a loud television misclassified as a scream) should not, by itself, be permitted to trigger an autonomous response. Second, classical evidence-combination frameworks such as Dempster-Shafer theory combine belief masses through an explicit combination rule that becomes computationally and conceptually unwieldy as the number of independent sources grows and requires each source to report a full belief structure rather than a scalar confidence [27]; JCFE instead operates on scalar per-module confidences, which keeps the fusion computation lightweight enough for continuous on-device execution while still capturing the essential benefit of multi-source corroboration through A(t). Third, generic late-fusion architectures drawn from the multimodal machine learning literature typically assume a fixed-dimensional input across all instances [22]; JCFE is instead defined over a variable-length active module set M, since the set of modules producing output changes from instant to instant depending on whether speech, motion, or location evidence is currently available. This combination of scalar-confidence corroboration, temporal persistence, and variable-arity operation is what distinguishes JCFE from both generic weighted fusion and classical evidence theory, and is the specific technical property that makes it suitable for continuous, on-device, multimodal emergency inference rather than a direct application of either prior approach.

Two consistency properties follow directly from the construction of Eq. (4). First, boundedness: because σ(·) maps the real line onto the open interval (0, 1), E(t) is guaranteed to lie in (0, 1) regardless of the magnitude of its argument, so no combination of module confidences can produce a probability outside the valid range. Second, monotonicity: since R(t), A(t), and P(t) each enter Eq. (4) with a non-negative coefficient and σ(·) is a strictly increasing function, E(t) is non-decreasing in each of R(t), A(t), and P(t) individually, meaning that all else equal, additional corroborating evidence, stronger inter-module agreement, or more sustained evidence can only increase, never decrease, the fused emergency probability. This monotonicity is a desirable and explicitly verified property for a safety-critical fusion function, since it guarantees that JCFE cannot exhibit counter-intuitive behavior in which stronger evidence lowers the system's assessed risk.

E. Adaptive Decision Threshold

Rather than comparing E(t) against a single fixed threshold, the Adaptive Emergency Decision Engine compares it against a threshold τ(t) that itself adapts to context and to the system's recent false-alarm history:

$$\tau(t) = \tau 0 - \lambda 1 \cdot h(Z(t)) + \lambda 2 \cdot F(t) \quad (5)$$

where τ0 is a baseline threshold established during offline validation; h(Z(t)) ∈ [0, 1] lowers the effective threshold in contexts independently associated with elevated risk, such as unfamiliar locations at unusual hours; F(t) raises the threshold in proportion to the system's recent false-alarm rate, so that a device experiencing frequent false positives becomes conservatively less sensitive until confidence in its calibration is restored; and λ1, λ2 are non-negative tuning coefficients.

F. Decision Function

The final autonomous response decision D(t) ∈ {0, 1} is given by:

$$D(t) = 1 \ \ if\ E(t) \geq \tau(t),\ \ else\ \ D(t) = 0 \quad (6)$$

When D(t) = 1, the Response Layer executes the autonomous emergency response pipeline described in Section V-E. When D(t) = 0 but E(t) exceeds a lower, secondary threshold τwatch(t) < τ(t), the system enters a

heightened-monitoring state in which sampling rates are increased and the persistence term P(t) is given additional weight in subsequent evaluations, allowing gradually accumulating evidence to eventually cross τ(t) without requiring any single instant of overwhelming confidence.

G. False-Alarm Reduction Formalization

The false-alarm rate over an evaluation window of N decision instants is defined in the standard way as the proportion of positive decisions, D(t) = 1, that do not correspond to a genuine emergency:

$$FAR = (\Sigma t\ 1[D(t) = 1 \wedge \neg Emergency(t)]) \ / \ (\Sigma t\ 1[D(t) = 1]) \quad (7)$$

The agreement term A(t) and persistence term P(t) in Eq. (4), together with the false-alarm feedback term F(t) in Eq. (5), are the three principal mechanisms by which JCFE and the adaptive decision engine jointly suppress FAR: requiring either multi-module corroboration or sustained evidence before triggering, and adaptively raising the threshold in response to recent false-positive history, rather than requiring any single module to independently achieve an unrealistically low false-positive rate on its own.

VIII. ALGORITHMS

This section presents pseudocode for the two algorithms central to the proposed framework: the Javipra Confidence Fusion Engine (Algorithm 1) and the Adaptive Emergency Decision procedure that governs the continuous monitoring loop (Algorithm 2).

Algorithm 1: Javipra Confidence Fusion Engine (JCFE)

```
Input: active module set M = {m1,...,mk};
       confidence scores {ci}; base reliabilities
{ri};
       context Z; recent risk history buffer H
Output: Emergency Probability E

1: for each mi in M do
2:     wi <- ri * g(Z, mi)          // contextual
weight, Eq.(2)
3: end for
4: R <- (Σ wi*ci) / (Σ wi)          // risk score,
Eq.(3)
5: A <- 1 - normalizedVariance({ci}) // inter-
module agreement
6: append R to H; trim H to window W
7: P <- exponentialMovingAverage(H)  // persistence
term
8: E <- sigmoid(alpha*R + beta*A + gamma*P - delta)
// Eq.(4)
9: return E
```

Algorithm 2: Adaptive Emergency Decision and Response Loop

```
Input: continuous sensor streams; baseline
threshold tau0;
       false-alarm history F
Output: autonomous response actions (if triggered)

1: state <- NORMAL
2: loop forever
3:     acquire latest sensor frames (audio, IMU,
GPS)
4:     M <- run active Perception Layer modules
5:     Z <- computeContext(time, location,
activityHistory)
6:     E <- JCFE(M, Z, H)                        //
Algorithm 1
7:     tau <- tau0 - lambda1*h(Z) + lambda2*F
// Eq.(5)
8:     if E >= tau then
9:         triggerEmergencyResponse()
10:            shareLiveLocation()
11:            notifyTrustedContacts()
12:            beginEncryptedEvidenceCapture()
13:        state <- ALERT
14:    else if E >= tau_watch then
15:        state <- HEIGHTENED_MONITORING
16:        increaseSamplingRate()
17:    else
18:        state <- NORMAL
19:    end if
20:    updateFalseAlarmHistory(F, state,
userFeedback)
21: end loop
```

IX. IMPLEMENTATION

This section describes the proposed implementation stack for Javipra AI. Because the present paper reports the framework's design and mathematical formulation rather than a deployed evaluation, the implementation is described at the level of selected technologies and the rationale for each choice, consistent with the experimental methodology presented in Section X.

TABLE 2

Proposed Implementation Technology Stack

Layer	Technology	Rationale
Mobile application shell	Flutter	Cross-platform deployment from a single codebase, native-like performance, and mature plugin support for sensor and background-service access.
On-device inference	TensorFlow Lite	Purpose-built for quantized, low-latency inference on mobile hardware, with broad operator support for the convolutional and recurrent architectures used by the Perception Layer modules.
Perception module development	Python (TensorFlow / Keras)	Standard, well-supported environment for training and quantizing the HAR, speech, and audio models prior to conversion to TensorFlow Lite format.
Sensor access	Android Sensor APIs	Native access to accelerometer, gyroscope, and microphone streams with configurable sampling rates and background execution.
Location services	Platform GPS APIs	Access to device location with adjustable update frequency, enabling the duty-cycled polling strategy described in Section V-A.

Trusted-contact notification and evidence sync	Firebase (Cloud Messaging and Storage)	Reliable, low-latency delivery of emergency alerts to trusted contacts and optional encrypted upload of evidence once an emergency is confirmed, without requiring raw sensor data to be transmitted under normal operation.
On-device encryption	Platform Keystore-backed symmetric encryption	Protects locally stored emergency audio and evidence recordings at rest, consistent with the privacy-by-architecture principle of Section IV.

A. Rationale for Edge AI Execution

TensorFlow Lite was selected specifically because it supports post-training quantization and operator fusion, techniques shown in the TinyML and efficient-inference literature to substantially reduce model size and inference latency with modest accuracy cost [15]–[17], [19]. Following the model-compression practices established in this literature, each Perception Layer model is targeted for int8 post-training quantization, which typically reduces model size by a factor of approximately four relative to a float32 baseline with limited accuracy degradation [15], [19]. The Voice Keyword Spotting and Human Activity Recognition modules, being the most latency-sensitive, are targeted at compact convolutional architectures of the kind shown to fit within tens to a few hundred kilobytes after quantization in the small-footprint keyword-spotting literature [11], [12], while the Speech Emotion Recognition and Audio Event Detection modules, invoked only when speech or acoustic events are present rather than continuously, can accommodate a moderately larger footprint. This allows the five Perception Layer modules described in Section V-B to execute concurrently within the memory and power budget of a typical smartphone, without dependence on network connectivity, directly addressing the latency and offline-availability motivations discussed in Section I.

Concurrency is managed by executing the audio-derived modules (KWS, SER, AED) as a single batched inference pass over a shared short-time audio buffer where architecturally feasible, since all three consume the same input stream, while the HAR module runs on an independent, continuously updated inertial buffer at a shorter inference interval appropriate to motion dynamics. The GPS Context Analysis module executes least frequently, consistent with the lower semantic bandwidth of location updates relative to audio and inertial signals. This staggered, workload-appropriate scheduling is intended to keep aggregate CPU and battery utilization within the bounds characterized for comparable concurrent on-device inference workloads in the TinyML literature [15]–[17], though, consistent with Section XII, the specific utilization figures for the integrated five-module pipeline remain to be measured empirically rather than assumed.

B. Rationale for Flutter as the Application Shell

Flutter was selected over platform-specific native development to allow a single codebase to target both Android and iOS, reducing the engineering burden of maintaining parity between platforms for a safety-critical application, while its plugin ecosystem provides the background execution and sensor access required by the continuous monitoring workflow described in Section VI.

C. Data Handling and Privacy Implementation

Consistent with the privacy-preserving design principle established in Section IV, the implementation stores raw audio and inertial buffers only transiently in memory, discarding each buffer once the corresponding Perception Layer module has produced its confidence score. Only in the event of a confirmed emergency, when D(t) = 1 in Eq. (6), is any audio retained, and even then it is retained in encrypted form on-device, with transmission to trusted contacts limited to location, timestamp, and a brief system-generated summary rather than the raw recording itself, as depicted in the data flow of Fig. 5.

X. EXPERIMENTAL METHODOLOGY

As this paper presents the design, architecture, and mathematical formulation of Javipra AI, no experimental results are reported at this stage; fabricating performance figures would misrepresent the current state of the work. This section instead specifies the methodology by which the framework is intended to be validated in subsequent work, following the experimental design conventions established in the reviewed HAR, SER, and fall-detection literature [1]–[5], [7]–[9], [20], [21].

A. Evaluation Metrics

Consistent with standard practice in the emergency-detection and activity-recognition literature, evaluation is planned along four dimensions: detection performance, measured via precision, recall, F1-score, and the area under the receiver operating characteristic curve (AUC-ROC) for the binary decision D(t); false-alarm rate, measured as defined in Eq. (7) over extended periods of normal daily activity; response latency, measured as the elapsed time between the onset of emergency-consistent evidence and the triggering of D(t) = 1; and on-device resource consumption, measured as inference latency, peak memory usage, and battery drain per hour of continuous monitoring.

B. Testing Strategy

A three-stage testing strategy is proposed. Unit-level testing will evaluate each Perception Layer module independently against established modality-specific benchmark datasets, for example, public HAR datasets for the activity recognition module and public speech-emotion corpora for the SER module, to confirm that each module's standalone accuracy is consistent with the results reported in the literature reviewed in Section II before it is integrated into the fusion pipeline. Integration-level testing will evaluate the Javipra Confidence Fusion Engine and the adaptive decision engine using scripted, simulated scenarios that combine multiple modalities, both scenarios constructed to resemble genuine emergencies (e.g., a distress keyword co-occurring with a fall-consistent motion

signature) and scenarios constructed to resemble superficially similar but benign activity (e.g., vigorous exercise with loud ambient music), to evaluate whether JCFE correctly discriminates between them. Field-level testing, conducted only with informed consent and under institutional ethical oversight, will evaluate the complete system under naturalistic daily use to measure real-world false-alarm rate and battery impact over extended periods.

C. Suitable Datasets

Module-level validation can draw on existing public benchmarks: smartphone-based HAR datasets of the kind used in the surveys of Gu et al. and Ronao and Cho for the activity recognition and fall-detection modules [1], [4]; public speech-emotion corpora of the kind used in the SER literature reviewed in Section II-C for the emotion recognition module [7]–[9]; and publicly available acoustic-event datasets containing scream, shout, and impulsive-sound recordings, of the kind used by Valenzise et al., for the audio event detection module [10]. Because no public dataset combines synchronized audio, inertial, and location streams specifically labeled for emergency versus non-emergency ground truth, an integration-level dataset would need to be constructed via consented, scripted scenario recording, an activity planned for future work as discussed in Section XIII, rather than claimed as already completed.

D. Ablation Studies

To isolate the specific contribution of JCFE's design choices, the validation plan includes a structured ablation study rather than only an end-to-end accuracy measurement. Four ablated variants are proposed for comparison against the complete system: (i) JCFE without the agreement term ($\beta = 0$ in Eq. (4)), to quantify the benefit of rewarding inter-module corroboration; (ii) JCFE without the persistence term ($\gamma = 0$), to quantify the benefit of requiring sustained rather than instantaneous evidence; (iii) JCFE with a fixed rather than adaptive threshold ($\lambda 1 = \lambda 2 = 0$ in Eq. (5)), to quantify the benefit of context- and history-sensitive decision-making; and (iv) a single-modality baseline in which only the Human Activity Recognition module drives the decision, representing the most capable individual module in isolation. Comparing detection performance and false-alarm rate across these variants, using the metrics defined in Section X-A, would directly demonstrate, or refute, the marginal contribution of each architectural component introduced in Section VII, rather than presenting JCFE's advantage as an unverified assumption.

E. Baseline Comparisons

Beyond internal ablation, the validation plan proposes comparison against three external baseline categories. The first baseline category is manual-activation systems representative of the reviewed women's-safety literature [25], evaluated on response latency and on their inability to trigger at all in simulated incapacitation scenarios, to quantify concretely the coverage gap identified in Section III. The second baseline category is a naive fusion rule, specifically an unweighted logical OR across per-module binary decisions (i.e., an alert whenever any single module exceeds its own decision threshold), representing the simplest plausible multimodal alternative to JCFE and expected to exhibit a substantially higher false-alarm rate. The third baseline category is a conventional weighted-average fusion rule, $E(t) = R(t)$ from Eq. (3) without the agreement or persistence terms, isolating the specific benefit of the JCFE extensions discussed in Section VII-D over standard confidence averaging. Reporting precision, recall, F1-score, and false-alarm rate for JCFE alongside all three baselines, on the same evaluation scenarios, would constitute direct evidence for the framework's central claim of improved robustness over both manual and conventional automated alternatives.

F. Future Validation Plan

The validation plan proceeds in three phases: offline validation of each Perception Layer module against public benchmarks, together with the ablation and baseline comparisons of Sections X-D and X-E; simulated integration testing of JCFE and the decision engine using scripted multimodal scenarios; and, subject to institutional ethical review and informed consent, a limited-scale, consented field pilot to measure real-world false-alarm rate, response latency, and battery impact prior to any broader deployment claim.

XI. DISCUSSION

The design presented in this paper directly targets the structural failure mode identified in Section I: dependence on manual activation. By treating emergency detection as an autonomous, multimodal inference problem rather than a user-initiated event, Javipra AI is, in principle, capable of responding to the categories of emergency, assault, restraint, medical incapacitation, and panic, in which existing manual-trigger systems are least effective. The confidence-weighted design of JCFE, in particular the agreement and persistence terms in Eq. (4), reflects a deliberate design stance: individual modules are expected to be imperfect and occasionally wrong, and robustness is sought through principled combination rather than through demanding unrealistic accuracy from any single module.

The architecture also reflects an explicit prioritization of privacy alongside protection. Because every Perception Layer module executes on-device and raw sensor buffers are discarded immediately after inference, the system's continuous listening and motion-tracking capability does not, by architecture, constitute continuous data collection in the sense that would raise the surveillance concerns discussed in Section I. This positions Javipra AI's privacy posture as substantially different from cloud-dependent alternatives, in which continuous audio or location streaming would be required for equivalent functionality.

At the same time, the discussion in Section III's gap analysis and the honest limitations enumerated in Section XII should be read together: the architectural design proposed here addresses the structural gaps identified in the literature, but its practical effectiveness, in particular its true positive and false alarm rates under naturalistic conditions, can only be established through the staged validation methodology described in

Section X, which has not yet been executed at the time of this writing.

The ablation and baseline structure proposed in Sections X-D and X-E is itself part of the technical discussion this paper aims to enable: because JCFE's claimed advantage over conventional weighted-average fusion rests specifically on the agreement term A(t) and persistence term P(t) in Eq. (4), the corresponding ablated variants are designed so that any measured improvement can be attributed to a specific mechanism rather than to the framework as an undifferentiated whole. Similarly, comparing JCFE against a naive logical-OR fusion baseline is intended to make explicit a trade-off inherent to any multimodal safety system: a logical-OR rule maximizes recall at the cost of an expected high false-alarm rate, while JCFE's corroboration- and persistence-aware design is intended to retain high recall for sustained or multiply-corroborated evidence while suppressing the isolated single-module triggers most likely to be false positives. Whether this intended trade-off is realized in practice, and to what quantitative degree, is precisely the question the proposed baseline comparison is designed to answer, and the present paper is careful not to presuppose the answer in advance of that evaluation.

XII. LIMITATIONS

Several limitations of the present work should be acknowledged candidly. First, this paper presents an architectural and mathematical framework together with a proposed implementation stack; it does not yet report empirical performance results, since no integrated prototype has been evaluated under the methodology of Section X. Consequently, the true detection accuracy, false-alarm rate, and battery impact of the complete system remain to be established rather than assumed. Second, the free parameters of the mathematical model, specifically α, β, γ, δ in Eq. (4), $\lambda 1$, $\lambda 2$ in Eq. (5), and the base reliabilities r_i in Eq. (2), require calibration against labeled data before deployment; the present paper defines their role formally but does not claim specific fitted values. Third, continuous multimodal sensing, even when executed entirely on-device, carries an inherent battery and thermal cost that must be empirically characterized and may require further optimization, such as more aggressive duty-cycling of the audio pipeline, before the system is suitable for all-day use. Fourth, acoustic and motion-based modules are subject to well-known real-world confounds, ambient noise, multiple simultaneous speakers, or unusual but benign motion patterns, that can degrade module-level confidence estimates in ways that context-layer adjustment can only partially mitigate. Fifth, the response pipeline depends on the availability of network connectivity for trusted-contact notification and, where applicable, live-location sharing; while on-device detection itself is designed to function offline, the final autonomous response step is necessarily constrained by prevailing network conditions at the moment an emergency is detected. Sixth, any autonomous safety system carries an ethical responsibility to minimize both false negatives and false alarms, and the consequences of miscalibration in either direction, a missed genuine emergency or a disruptive false alert, are non-trivial and must be weighed carefully during the field-validation phase proposed in Section X.

XIII. FUTURE WORK

Future work proceeds directly from the limitations enumerated in Section XII. The immediate next step is the construction of an integrated prototype implementing the architecture of Section V using the technology stack of Section IX, followed by the staged validation methodology of Section X, including the ablation studies and baseline comparisons of Sections X-D and X-E, beginning with module-level benchmarking, proceeding to simulated multimodal integration testing, and culminating in a consented, ethically reviewed field pilot. In parallel, the free parameters of the mathematical model introduced in Section VII should be calibrated empirically, using labeled scenario data collected during integration testing, rather than fixed heuristically.

Beyond immediate validation, several further directions follow from the architecture presented in this paper. First, the Perception Layer could be extended with additional modalities, such as heart-rate or other wearable-derived physiological signals where available, to further strengthen the evidentiary basis for fusion, provided that any new modality is integrated through the same variable-arity, scalar-confidence interface that allows JCFE to remain agnostic to the number and type of active modules. Second, on-device federated learning, following the techniques surveyed by Kairouz et al. and Li et al. [23], [24], offers a path for JCFE's weighting coefficients and per-module reliability estimates r_i to improve over time from aggregated, privacy-preserving feedback across a population of devices, without centralizing raw data at any point, an approach directly compatible with the privacy-by-architecture principle of Section IV. Third, a formal user study of trusted-contact response behavior and false-alarm tolerance would allow the adaptive threshold of Eq. (5) to be calibrated against real human factors, including the psychological and social cost of false alerts to trusted contacts, rather than engineering assumptions alone. Fourth, the false-alarm feedback term $F(t)$ in Eq. (5) is currently defined at the level of an individual device's own history; an important extension is to investigate whether population-level false-alarm statistics, aggregated through the federated mechanism above, can provide a more stable and faster-converging threshold-adaptation signal than a single device's limited local history. Fifth, formal verification techniques could be applied to the decision function of Eq. (6) and its monotonicity property established in Section VII-D, to provide certified guarantees about JCFE's worst-case behavior under adversarial or corrupted module outputs, which is particularly relevant given that Section XII identifies acoustic and motion confounds as an open robustness concern. Sixth, subject to further regulatory and ethical review, integration with formal emergency-service dispatch systems, rather than trusted-contact notification alone, represents a natural extension of the Response Layer once the false-alarm rate has been rigorously characterized through the validation plan of Section X.

XIV. CONCLUSION

This paper has presented Javipra AI, a confidence-weighted multimodal sensor fusion framework for autonomous, privacy-preserving emergency detection and response on edge devices, motivated by the structural inadequacy of manually activated emergency-response applications in exactly the categories of danger, assault, kidnapping, medical incapacitation, falls, unconsciousness, restraint, and panic, where existing systems are least effective. Building on established research in human activity recognition, speech emotion recognition, audio event detection, keyword spotting, multimodal sensor fusion, and edge and tiny machine learning, this work identified five specific gaps that persist when these strands are considered jointly, and proposed the Javipra Confidence Fusion Engine as a domain-specific mechanism for combining heterogeneous, independently produced AI outputs into a single, calibrated emergency probability, together with an adaptive decision engine designed explicitly to reduce false alarms without sacrificing sensitivity to genuine danger. Unlike conventional weighted-average fusion or classical evidence-combination frameworks, JCFE explicitly models inter-module agreement and temporal persistence as first-class evidentiary signals while operating over a variable-length set of active modules, a combination of properties shown in Section VII-D to be necessary for continuous, on-device, multimodal emergency inference. The system architecture, mathematical formulation, and algorithmic realization presented in Sections V through VIII, illustrated through seven publication-quality architectural, workflow, and interaction diagrams, establish a complete design for the framework, while the proposed implementation stack and experimental methodology of Sections IX and X, including concrete ablation studies and baseline comparisons, chart a concrete path toward empirical validation. Consistent with the limitations acknowledged in Section XII, this paper's contribution is the framework and its formal foundation rather than a validated deployment; the future work outlined in Section XIII defines the steps required to move from this design to a rigorously evaluated system capable of providing autonomous protection precisely in the moments when existing safety technology offers the least.

REFERENCES

[1] F. Gu, M.-H. Chung, M. Chignell, S. Valaee, B. Zhou, and X. Liu, "A survey on deep learning for human activity recognition," ACM Computing Surveys, vol. 54, no. 8, pp. 1–34, 2021.

[2] F. Demrozi, G. Pravadelli, A. Bihorac, and P. Rashidi, "Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey," IEEE Access, vol. 8, pp. 210816–210836, 2020.

[3] E. Ramanujam, T. Perumal, and S. Padmavathi, "Human activity recognition with smartphone and wearable sensors using deep learning techniques: A review," IEEE Sensors Journal, vol. 21, no. 12, pp. 13029–13040, 2021.

[4] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," Expert Systems with Applications, vol. 59, pp. 235–244, 2016.

[5] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architecture for human activity recognition," IEEE Access, vol. 8, pp. 56855–56866, 2020.

[6] F. J. Dian, R. Vahidnia, and A. Rahmati, "Wearables and the Internet of Things (IoT), applications, opportunities, and challenges: A survey," IEEE Access, vol. 8, pp. 69200–69211, 2020.

[7] D. Issa, M. F. Demirci, and A. Yazici, "Speech emotion recognition with deep convolutional neural networks," Biomedical Signal Processing and Control, vol. 59, p. 101894, 2020.

[8] M. B. Er, "A novel approach for classification of speech emotions based on deep and acoustic features," IEEE Access, vol. 8, pp. 221640–221653, 2020.

[9] A. Mukhamediya, S. Fazli, and A. Zollanvari, "On the effect of log-mel spectrogram parameter tuning for deep learning-based speech emotion recognition," IEEE Access, vol. 11, pp. 61950–61957, 2023.

[10] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in Proc. 2007 IEEE Conf. Advanced Video and Signal Based Surveillance (AVSS), London, U.K., 2007, pp. 21–26.

[11] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in Proc. 2014 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 2014, pp. 4087–4091.

[12] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," arXiv preprint arXiv:1711.07128, 2017.

[13] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in Proc. INTERSPEECH, 2015, pp. 1478–1482.

[14] I. López-Espejo, Z.-H. Tan, J. H. L. Hansen, and J. Jensen, "Deep spoken keyword spotting: An overview," IEEE Access, vol. 10, pp. 4169–4199, 2022.

[15] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, "A comprehensive survey on TinyML," IEEE Access, vol. 11, pp. 96892–96922, 2023.

[16] L. Capogrosso, F. Cunico, D. S. Cheng, F. Fummi, and M. Cristani, "A machine learning-oriented survey on tiny machine learning," IEEE Access, vol. 12, pp. 23406–23426, 2024.

[17] S. S. Saha, S. S. Sandha, and M. Srivastava, "Machine learning for microcontroller-class hardware: A review," IEEE Sensors Journal, vol. 22, no. 22, pp. 21362–21390, 2022.

[18] J. Chen and X. Ran, "Deep learning with edge computing: A review," Proceedings of the IEEE, vol. 107, no. 8, pp. 1655–1674, 2019.

[19] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," Proceedings of the IEEE, vol. 105, no. 12, pp. 2295–2329, 2017.

[20] A. Z. Rakhman, L. E. Nugroho, Widyawan, and Kurnianingsih, "Fall detection system using accelerometer and gyroscope based on smartphone," in Proc. 2014 1st Int. Conf. Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, Indonesia, 2014, pp. 99–104.

[21] F. De Cillis, F. De Simio, F. Guido, R. Antonelli Incalzi, and R. Setola, "Fall-detection solution for mobile platforms using accelerometer and gyroscope data," in Proc. 37th Annu. Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, 2015, pp. 3727–3730.

[22] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," IEEE Transactions on

Pattern Analysis and Machine Intelligence, vol. 41, no. 2, pp. 423–443, 2019.

[23] P. Kairouz et al., "Advances and open problems in federated learning," Foundations and Trends in Machine Learning, vol. 14, no. 1–2, pp. 1–210, 2021.

[24] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50–60, 2020.

[25] A. V. Agarwal, V. Singh, A. Kamboj, A. Sirohi, and A. Mehto, "Development of a women safety smartphone application—SAKHI," in Proc. 2023 3rd Int. Conf. Secure Cyber Computing and Communication (ICSCCC), 2023.

[26] S. Qiu, H. Zhao, N. Jiang, Z. Wang, L. Liu, Y. An, H. Zhao, X. Miao, R. Liu, and G. Fortino, "Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges," Information Fusion, vol. 80, 2022.

[27] G. Shafer, A Mathematical Theory of Evidence. Princeton, NJ, USA: Princeton University Press, 1976.

