

AI-Driven Safety and Health Monitoring Systems: A Review of Adaptive Learning, Multimodal Sensor Fusion, and Predictive Intelligence

Mrs. Vetrikani R
Asst. Prof., CSE (AI & ML)
EPCET
Bengaluru, India
vetrikaniir.eee@eastpoint.ac.in

Jeevika B S
Student, CSE (AI & ML)
EPCET
Bengaluru, India
sureshasha1999@gmail.com

Muskaan
Student, CSE (AI & ML)
EPCET
Bengaluru, India
muskaansheikh2212@gmail.com

Tejashwini R
Student, CSE (AI & ML)
EPCET
Bengaluru, India
tejashwiniramurthy@gmail.com

Mythri B K
Student, CSE (AI & ML)
EPCET
Bengaluru, India
mythribk2005@gmail.com

Prajwal S Ramaga
Student, CSE (AI & ML)
EPCET
Bengaluru, India
prajwalrahul4@gmail.com

Abstract

Research on safety and health monitoring includes wearable fall detection, human activity recognition, computer vision, connected sensors, and chronic-disease monitoring. These areas share an interest in identifying events that may require assistance, but results from an individual dataset or task do not demonstrate that a complete emergency-response system will work reliably. This paper describes LENORA, a proposed AI-assisted monitoring system covering child safety, women's safety, elderly monitoring, and adult disease-care monitoring. Its intended workflow links category-specific inputs with event analysis, risk assessment, dashboard support, safe-word verification, and location-aware communication. The literature summaries supplied for this review highlight practical concerns such as class imbalance, privacy, energy use, and deployment constraints. Project notes estimate that the software foundation is approximately 80% complete; this is a progress estimate, not a model-performance result. The available material does not verify that all sensors, AI models, GPS, GSM/SIM800L communication, and emergency-response steps have been integrated and tested. For that reason, this paper reports no LENORA accuracy, precision, recall, F1-score, latency, or deployment results. Instead, it records the current evidence boundaries and outlines tests for classification quality, false alarms, response time, communication reliability, and end-to-end behavior.

Keywords—wearable sensing, fall detection, Internet of Things, edge AI, risk assessment, emergency alerting, health monitoring

1. Introduction

Automated monitoring systems are intended to recognize events or changes that may call for assistance, including situations in which a person cannot place a call or press an SOS control. The literature considered here addresses falls among older adults, human activity recognition, computer-vision monitoring, safety applications for women and children, and chronic-disease monitoring [1]–[11]. These problems differ in their sensors, target events, users, and validation requirements. A model that classifies movement from inertial data, for example, does not by itself show that a system can interpret physiological measurements, determine

a location, deliver an SMS, or complete an emergency workflow.

LENORA is being developed around four intended monitoring categories: child safety, women's safety, elderly monitoring, and adult disease-care monitoring. The proposed architecture brings together sensor inputs, event analysis, risk assessment, user verification, GPS, GSM/SIM800L communication, and a dashboard. Each feature must be described according to the evidence available for its current implementation, rather than according to the final design goal.

A. Problem Statement

Much of the reviewed work evaluates a particular model or monitoring task. A practical response system, however, depends on several connected stages: acquiring usable input, checking data quality, making an inference, applying decision rules, interacting with the user, obtaining location when needed, communicating an alert, and recording the event. A failure at any stage can prevent the system from providing useful assistance. LENORA therefore needs a documented architecture as well as a clear separation between software that has been implemented and integrations that have not yet been verified.

B. Objectives and Contributions

This paper has four aims. First, it organizes the supplied literature summaries around topics relevant to LENORA. Second, it describes the proposed four-category architecture and the intended path from an event to an alert. Third, it separates the reported software progress from model and hardware performance that has not been verified. Finally, it sets out tests that should be completed before quantitative claims are made. The paper is therefore a design and implementation-status report; it does not claim validated emergency-response or medical performance.

C. Scope and Organization

The review is limited to the studies selected in the supplied draft; no new systematic literature search was conducted for this paper. Section II explains the review basis, Section III groups the literature by theme, and Sections IV and V describe the proposed architecture and methodology. Section VI records the implementation status available in the project notes. Section VII outlines an evaluation plan, and Sections

VIII–X discuss gaps, limitations, and future work before the conclusion and references.

II. Literature Review Methodology

The source literature-review manuscript summarizes papers on the SisFall dataset, wearable fall detection and risk assessment, human activity recognition, IoT-based fall detection, computer-vision activity recognition and fall detection, safety systems for women and children, and IoT-based chronic-disease monitoring. It also included a paper on live-birth potential in mouse embryos. The source draft left that paper out of the thematic comparison because its application area differs from the monitoring problems addressed by LENORA.

The material supplied for this paper did not include a reproducible database-search log, search date, full search query, or independent screening record. The review is consequently presented as a synthesis of preselected studies, not as a systematic review. Any review methods described in the original studies remain attributed to those studies; they are not presented as procedures carried out for this manuscript.

A. Evidence and Attribution Rules

Performance figures reported in the cited papers apply to those studies' datasets, participants, and test protocols. They are not LENORA results. Before submission, bibliographic details should be checked against publisher records, particularly for references whose venue, volume, page range, author list, or DOI information is incomplete in the supplied draft.

III. Technical Background and Thematic Review

A. Wearable and Inertial Sensing

Wearable fall-detection and activity-recognition systems often use accelerometers, gyroscopes, or other inertial measurement units. The supplied summary of SisFall describes a waist-worn device and recordings of everyday activities alongside simulated falls [1]. The FD-DNN paper examines a deep-learning method and reports results from its own offline and online test conditions [2]. Because sensor placement, sampling frequency, window length, and event definitions can differ between studies, their reported results should not be compared without considering those differences.

B. Machine Learning and Deep Learning

The supplied summaries discuss conventional machine-learning methods, including support vector machines and Naive Bayes, as well as CNNs, LSTMs, recurrent and residual networks, and ensemble approaches [2]–[4], [8]. CNNs can learn local patterns in input signals, while recurrent models can represent changes across a sequence. These are general properties of the methods discussed in the literature; they do not mean that the same models have been trained or deployed in LENORA. Class imbalance also matters: emergency events may be much less common than routine activities, so overall accuracy alone may hide missed emergencies.

C. Human Activity Recognition and Fall-Risk Assessment

The reviewed work on wearable fall-risk assessment and human activity recognition considers issues such as sensor placement, feature representation, model selection, battery life, comfort, and privacy [5], [6]. The activity-recognition toolbox in [7] has a related but narrower purpose: classifying everyday movements from accelerometer data. Recognizing an activity, detecting a fall, estimating risk, and sending an emergency alert are distinct tasks. Each therefore needs its own definition of success and its own tests.

D. IoT and Computer-Vision Monitoring

The IoT fall-detection surveys describe architectures that connect sensors, data processing, classification, and alert delivery [10]. The supplied study on women's and children's safety reports an IoT approach evaluated with its own data and machine-learning methods [8]. The computer-vision review considers RGB, depth, infrared, and skeleton-based representations, alongside challenges such as occlusion, lighting, privacy, and representation of intended users [9]. These sources help identify design choices and limitations, but they do not validate LENORA's implementation.

E. Chronic-Disease Monitoring

The supplied review of chronic-disease monitoring discusses IoT sensing and several machine-learning approaches for health-related applications [11]. In practice, such systems must account for measurement quality, user adherence, privacy, security, and clinical context. LENORA's health category should therefore be described as monitoring and decision support unless and until suitable validation supports stronger claims; it should not be presented as a diagnostic device.

F. Synthesis

Taken together, the supplied summaries show why performance must be interpreted in relation to the task, dataset, population, and test protocol. A useful system must also be assessed for false alarms, missed events, robustness, energy use, privacy, communication, and integration. For LENORA, this supports a modular design and component-by-component testing rather than a broad claim that one algorithm is best for every monitoring category.

IV. Proposed LENORA System Architecture

The proposed LENORA architecture separates user and profile information, category-specific inputs, data and API processing, event analysis, risk assessment, dashboard decision support, and the intended emergency-response functions. Figure 1 is a conceptual diagram drawn from the project documentation. It shows the intended sequence; it should not be read as evidence that every component or connection is already operational.

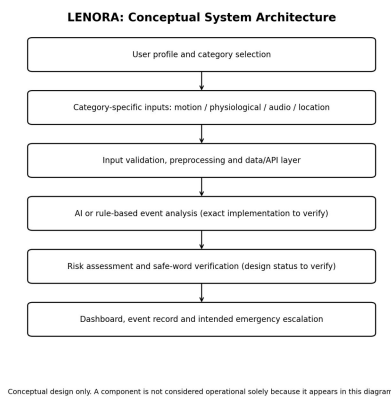


Fig. 1. Conceptual LENORA architecture. Components and connections require verification against the current source code and hardware.

A. Monitoring Categories

Category	Intended scope	Evidence/status boundary
Child safety	Movement/fall-related events, distress indicators, safe-zone context, and escalation.	Specific detectors and end-to-end workflow require implementation and testing evidence.
Women's safety	Distress indicators, contextual/location risk information, safe-word verification, and emergency support.	Crowd analysis is future scope; zone labels must not be presented as guarantees of safety.
Elderly monitoring	Fall/activity events, health context, medication records or reminders, and caregiver escalation.	Software foundations are described in project notes; sensor and alert operation remain to be verified.
Adult disease-care	Condition-related monitoring and health context; Parkinson's has been discussed as an initial focus.	No disease diagnosis or clinically validated model is established by the supplied evidence.

B. Input and Software Layers

Project notes list motion and physiological sensors, a microphone, GPS, SIM800L GSM communication, and an Arduino-class board among the planned or considered components. The final bill of materials and connected hardware configuration still need confirmation. The documented software foundation includes React/Vite and Node.js/Express. The actual API endpoints, database schema, trained models, and runtime behavior should be confirmed by examining and testing the project repository.

C. Offline-First Intent

One design goal is to keep essential functions from depending entirely on an internet connection or smartphone interaction, using local processing and GSM alerts where the hardware supports them. This remains an intended capability rather than a demonstrated result. An offline-first claim should be made only after tests disconnect internet access and record which functions continue to work, including whether GSM communication remains available.

V. Proposed Methodology

A. Input Acquisition and Preprocessing

The proposed pipeline begins with category-specific inputs and checks signal quality before analysis. For inertial signals, the implementation report should identify the sensor axes, sampling rate, window length, filtering, normalization, missing-data handling, and event labels. Physiological, audio,

and location inputs require their own processing steps and privacy controls. The material available for this manuscript does not establish LENORA's final sampling rate, dataset size, or complete preprocessing configuration.

B. Event Analysis

The event-analysis stage is intended to produce an event label and, where the implementation supports it, a confidence or severity indicator. The model family, trained weights, input features, class definitions, training data, and inference environment must be recorded from the actual implementation.

C. Risk Assessment

Project notes propose considering more than a single model prediction before escalating an event. Possible inputs include prediction confidence, event duration or severity, concurrent indicators, safe-zone context, health or activity history, and the outcome of safe-word verification. The exact formula, thresholds, handling of missing inputs, and escalation rules must be checked against executable code and tested. This paper does not assign numerical thresholds that have not been verified.

D. Safe-Word Verification and Alerting

The proposed workflow includes a local alert and, where applicable, a safe-word verification step. If an event is not cancelled, the intended sequence is to obtain a location, prepare a map link, send an SMS through GSM, and record the event. Project notes specify different safe-word policies for women and children. However, the available material does not verify the complete runtime sequence, including speech recognition or input handling, safe-word comparison, GPS acquisition, SMS delivery, and event logging.

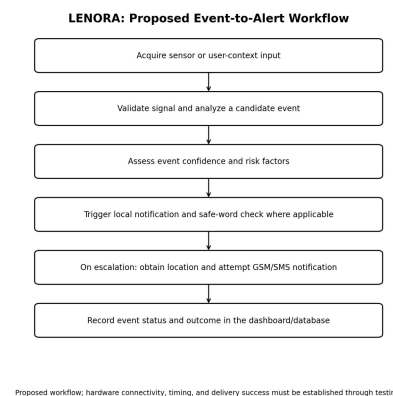


Fig. 2. Proposed event-to-alert workflow. Each stage requires a test case and recorded outcome before being reported as operational.

E. Privacy and Safety

LENORA may process location, health, audio, and emergency-contact information. The implementation should collect only what is necessary, limit access, protect credentials and safe-word data, and define consent, retention, and deletion practices. A model-generated alert is uncertain evidence; it must not be presented as a medical diagnosis or as a guarantee that a location is safe or unsafe.

VI. System Implementation and Current Status

Project notes describe a React/Vite frontend, a Node.js/Express backend, authentication and category navigation, category interfaces or placeholders, and components related to persistence. These descriptions reflect the recorded project status, not a fresh code audit or a successful end-to-end run. The repository, build output, model files, hardware test records, and complete workflow logs were not supplied with this manuscript, so the status table should be read with that limitation in mind.

Component	Current documented status	Publication wording / verification needed
Authentication and category navigation	Reported completed in project notes	Verify a clean run and functional navigation tests.
Frontend/backend foundation	React/Vite and Node.js/Express reported	Verify package versions, entry points, routes, and build.
Database and persistence	Described as implemented or scaffolded	Audit schema and test create/read/update/delete paths.
Category dashboards	Some interfaces/placeholders reported; deeper workflows in progress	Do not imply all monitoring logic is complete.
Risk assessment	Design or partial-stub status in project notes	Verify executable logic, thresholds, and unit tests.
AI models and training	Exact model, data, and experiment records not verified	Provide model files, training scripts, provenance, and held-out tests.
Sensor acquisition	Hardware planned/listed; live input not evidenced here	Provide wiring, firmware, and repeatable sensor logs.
Safe-word workflow	Design described; runtime unverified	Test valid/invalid inputs, cancellation, and timeout behavior.
GPS and SIM800L/GSM	Planned or unverified integration	Record location-fix outcomes and SMS delivery results.
Crowd analysis	Future scope; not trained according to project notes	Keep out of implemented-feature claims.
Overall software progress	Project-owner estimate: approximately 80%	Progress estimate only; not a quality or performance metric.

Status terminology used in this paper: “implemented” means code exists and passes a defined test; “partially implemented” identifies a working subset; “planned” means intended but not verified as implemented; and “unverified” means that adequate evidence was unavailable for this review.

VII. Proposed Evaluation Methodology

The supplied manuscript contains no new experimental results for LENORA. The steps below describe a proposed evaluation protocol; they are not tests reported as completed.

A. Reproducibility and Test Environment

For each run, record the repository commit, operating system, processor or hardware, sensor model and placement, firmware, library versions, dataset source and license, class definitions, preprocessing steps, model configuration, data splits, random seeds, and test scripts. Keep evaluation data separate from training and tuning data. For a public dataset, identify the exact subset used and retain the task definitions supplied with that dataset.

B. Proposed Test Scenarios

Test group	Proposed procedure	Evidence to retain
Software workflow	Test registration/login, category selection, profile management, event creation, dashboard display, and persistence.	Test cases, logs, screenshots, database records.

Event classification	Evaluate each defined class on held-out data with class distribution documented.	Dataset manifest, confusion matrix, per-class metrics.
False alarms	Test ordinary activities and confusing patterns; measure false alerts over a defined observation period.	Timestamped event log and ground-truth labels.
Risk logic	Test every risk band, missing/conflicting inputs, and safe-word outcomes.	Decision table and expected-versus-actual results.
GPS/GSM	Test valid/invalid GPS fixes, weak/no signal, SIM failure, repeated alert, and SMS receipt.	Coordinates, timestamps, delivery outcomes, failure logs.
Offline resilience	Disable internet and test which functions remain operational; test GSM separately.	Connectivity state and outcome for each function.
End-to-end	Run controlled scenarios from input through verification, location, alert, and event history.	Scenario IDs and timestamps for each stage.

C. Metrics

For a classifier that has been tested, report precision, recall or sensitivity, specificity, F1-score, and a confusion matrix. Include accuracy when it remains informative given the class distribution, and provide class-wise results; confidence intervals may be useful when the sample size allows them. System-level evaluation should separately report event-to-decision time, event-to-notification time, alert-delivery success, GPS-fix success, false alerts per hour or day, and end-to-end completion rate. Define the denominator and timing boundaries for each measure before testing. No LENORA-specific values are available yet.

D. Results Reporting

Add results only after reproducible tests have been run. Keep model metrics separate from hardware and communication measurements, and from usability observations. If a test cannot be completed, label it as not evaluated. Results quoted from the literature may be discussed as findings of the cited studies, but must not be described as measurements of LENORA.

VIII. Research Gaps and Technical Challenges

Gap/challenge	Evidence in supplied literature	Implication for LENORA
Representation of intended users	Summaries describe simulated falls, limited participant groups, and generalization concerns [1], [2], [5], [9].	Evaluate on appropriate populations and avoid claims beyond the tested data.
Class imbalance	The summaries identify imbalanced event classes and threshold adjustment approaches [3], [4], [8].	Report class-wise metrics and false-negative/false-positive behavior.
Real-world validation	Summaries distinguish offline evaluation from online or complete hardware/alert testing [2], [3], [8].	Test the whole event-to-alert chain, not only a classifier.
Privacy and security	Sensitive physiological, audio, and location information raise privacy and security issues [6], [8], [9], [11].	Apply data minimization, access control, secure storage, and consent policies.
Hardware and energy constraints	The literature discusses battery, computation, GPS energy, and deployment constraints [2], [6], [8], [9], [11].	Measure energy/runtime only on actual hardware and disclose the setup.
Cross-study comparability	Studies differ in tasks, datasets, populations, and reported metrics [2]–[5], [8], [9].	Avoid ranking methods using non-comparable metrics.

These gaps are derived from the supplied summaries and are not presented as an exhaustive assessment of the entire field.

IX. Discussion

The reviewed literature supports treating sensing, event recognition, decision logic, communication, and validation as separate parts of a monitoring system. LENORA’s proposed

risk-assessment and safe-word stages could provide checks before an alert is escalated, but their value depends on clearly defined rules and tested handling of invalid or missing inputs. A dashboard can present information and support decisions; its existence does not demonstrate that the sensors or prediction models are functioning correctly.

The four-category scope presents a substantial engineering challenge because each category involves different events, input data, and safety requirements. A practical next step is to freeze a version of the repository, verify one clearly defined use case, build a reproducible data pipeline, and measure false positives and false negatives before adding more modalities. The reported 80% software progress describes development status only; it does not measure accuracy, reliability, or readiness for deployment.

X. Limitations and Future Work

A. Limitations

This paper draws on the supplied literature review and project notes. It is not based on a source-code audit or a completed experimental study. Several functions remain at the design stage or lack evidence of integrated hardware operation. Some references also need bibliographic verification against the publisher's record. Finally, one dataset or model cannot establish validity across all four monitoring categories. Health, audio, and location data also require appropriate privacy, consent, security, and retention controls. LENORA should not be described as a certified medical device or as a substitute for emergency services or clinical judgment.

B. Future Work

Future work should begin with a versioned repository and a documented check of frontend execution, backend behavior, and data persistence. Sensors should then be connected and tested one at a time. The team should define a single baseline classification task, record data provenance, and run a held-out evaluation. Risk rules, safe-word interaction, GPS/GSM messaging, and offline behavior can then be checked through controlled end-to-end scenarios. Features such as crowd analysis should remain future work until appropriate data, privacy safeguards, and validation procedures are available.

XI. Conclusion

This paper describes LENORA's proposed design and documented implementation status alongside supplied research on wearable fall detection, activity recognition, IoT safety, computer vision, and chronic-disease monitoring. The proposed architecture brings together category-specific inputs, event analysis, risk assessment, dashboard support, user verification, and emergency communication. The available evidence supports discussion of the design and reported software foundation, but it does not establish trained-model performance or a fully integrated hardware alert workflow. The evaluation plan identifies the evidence needed to assess classification quality, false alarms, response timing, communication reliability, and end-to-end operation. Quantitative results have not been added because they must come from completed, reproducible experiments.

References

- [1] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, "SisFall: A Fall and Movement Dataset," *Sensors*, vol. 17, no. 1, Art. no. 198, 2017, doi: 10.3390/s17010198.
- [2] L. Liu, Y. Hou, J. He, J. Lungu, and R. Dong, "An Energy-Efficient Fall Detection Method Based on FD-DNN for Elderly People," *Sensors*, vol. 20, no. 15, Art. no. 4192, 2020, doi: 10.3390/s20154192.
- [3] Z. Mohammad, A. R. Anwary, M. F. Mridha, M. S. H. Shovon, and M. Vassallo, "An Enhanced Ensemble Deep Neural Network Approach for Elderly Fall Detection System Based on Wearable Sensors," *Sensors*, vol. 23, Art. no. 4774, 2023, doi: 10.3390/s23104774.
- [4] J. Zhang, J. Li, and W. Wang, "A Class-Imbalanced Deep Learning Fall Detection Algorithm Using Wearable Sensors," 2021. Publication venue and complete bibliographic metadata were not provided in the supplied draft; verify before submission.
- [5] M. Chen et al., "A Systematic Review of Wearable Sensor-Based Technologies for Fall Risk Assessment in Older Adults," *Sensors*, vol. 22, no. 18, Art. no. 6752, 2022, doi: 10.3390/s22186752. Verify full author list against publisher record.
- [6] F. Serpush, M. B. Menhaj, B. Masoumi, and B. Karasfi, "Wearable Sensor-Based Human Activity Recognition in the Smart Healthcare System," *Computational Intelligence and Neuroscience*, vol. 2022, Art. no. 1391906, 2022, doi: 10.1155/2022/1391906.
- [7] F. Wieland and C. Nigg, "A Trainable Open-Source Machine Learning Accelerometer Activity Recognition Toolbox: Deep Learning Approach," *JMIR AI*, vol. 2, e42337, 2023, doi: 10.2196/42337.
- [8] N. R. Wagh et al., "Evaluation of IoT-Based Smart Safety Systems for Women and Children Using Machine Learning Techniques," *Scientific Reports*, vol. 16, Art. no. 87, 2026, doi: 10.1038/s41598-025-29146-4. Verify author list and final publication metadata.
- [9] F. X. Gaya-Morey, C. Manresa-Yee, and J. M. Buades-Rubio, "Deep Learning for Computer Vision Based Activity Recognition and Fall Detection of the Elderly: A Systematic Review," *Applied Intelligence*, vol. 54, pp. 8982–9007, 2024, doi: 10.1007/s10489-024-05645-1.
- [10] M. E. Karar, H. I. Shehata, and O. Reyad, "A Survey of IoT-Based Fall Detection for Aiding Elderly Care: Sensors, Methods, Challenges and Future Trends," *Applied Sciences*, vol. 12, no. 7, Art. no. 3276, 2022, doi: 10.3390/app12073276.
- [11] Y. Liu and B. Wang, "Advanced Applications in Chronic Disease Monitoring Using IoT Mobile Sensing Device Data, Machine Learning Algorithms and Frame Theory: A Systematic Review," *Frontiers in Public Health*, vol. 13, Art. no. 1510456, 2025, doi: 10.3389/fpubh.2025.1510456.