International Journal of Current Science Research
www.drbgrpublications.in

Cloud-Based Healthcare Monitoring System using CNN-LSTM for Early Detection of Patient Health Risks

S.Brindha Devi

Assistant Professor, Department of Computer Science, Sri S.Ramasamy Naidu Memorial College, Sattur, Tamil Nadu, India

Email: brindha.sb@gmail.com

Abstract

Continuous monitoring of physiological parameters is increasingly important for patients who require observation beyond conventional hospital settings. Traditional monitoring approaches often depend on periodic measurements and manual review, which can delay recognition of abnormal patterns. This paper proposes a cloud-based healthcare monitoring system that combines Internet of Things (IoT) data acquisition, cloud storage, automated preprocessing, and a convolutional neural network–long short-term memory (CNN-LSTM) model for early health-risk assessment. The proposed framework accepts time-dependent physiological observations such as heart rate, oxygen saturation, body temperature, blood pressure, respiratory rate, and electrocardiographic signals when available. A cloud layer provides centralized ingestion, storage, access control, and dashboard connectivity, while the deep-learning layer learns local signal representations and temporal dependencies. The system is designed as a research framework rather than a clinically validated diagnostic device; therefore, no fabricated patient results or accuracy values are reported. A methodology for dataset preparation, windowing, normalization, model training, evaluation, alert generation, and deployment is defined. The paper also discusses data quality, privacy, latency, interoperability, false alarms, explainability, and the limitations of cloud-dependent healthcare applications. Public physiological repositories such as PhysioNet can be used in future experimental validation, while real clinical deployment would require appropriate governance, validation, and regulatory review. The

proposed architecture provides a practical foundation for scalable remote patient monitoring and deep-learning-assisted clinical decision support.

Keywords: Cloud Computing, Healthcare Monitoring, Internet of Things, Deep Learning, CNN-LSTM, Remote Patient Monitoring, Physiological Signals, Risk Prediction

Introduction

Digital health systems increasingly connect patients, clinicians, sensing devices, software services, and cloud infrastructure. Remote monitoring is one important component of this ecosystem because it enables physiological observations to be collected outside a conventional bedside environment. The World Health Organization describes remote monitoring as part of the broader digital-health and telehealth landscape, where services can support care at a distance (WHO, 2025).

The uploaded reference paper follows a practical engineering structure that begins with an abstract and introduction, reviews related work, identifies system challenges, explains common techniques, compares approaches, and closes with conclusions and references. The present paper adopts that organization while changing the application domain from smart parking to healthcare monitoring. In particular, the reference paper emphasizes centralized cloud architectures, automation, data persistence, real-time tracking, and privacy considerations as important system-level concerns.

Healthcare monitoring presents additional challenges because physiological measurements are noisy, patient conditions evolve over time, and incorrect alerts can have clinical consequences. A useful monitoring platform therefore needs more than data collection. It should validate incoming measurements, manage missing or abnormal values, preserve patient records securely, analyze temporal patterns, and communicate risk information in a way that supports not replaces professional clinical judgment.

Deep learning is attractive for this problem because physiological data can contain both short-term signal patterns and longer temporal dependencies. Convolutional layers can learn local representations from transformed or multichannel signals, while recurrent layers such as LSTM units can model sequential dependencies. The proposed CNN-LSTM architecture combines these complementary capabilities in a cloud-oriented workflow.

The main contribution of this paper is a complete conceptual architecture for a cloud-based healthcare monitoring platform with deep-learning-assisted risk assessment. The contribution is methodological: it defines system components, data flow, preprocessing, model structure, evaluation criteria, alert logic, and deployment considerations without claiming experimental clinical performance.

Related Work

Healthcare monitoring has evolved from manually recorded observations toward connected sensing and digital platforms. IoT-based healthcare systems generally combine sensors, communication networks, processing services, and user interfaces. Survey literature has described the use of IoT technologies for health data acquisition and remote care, while cloud computing provides scalable storage and computational services for distributed applications (Istepanian, *et al*. 2011, Islam, *et al*. 2015).

Cloud-based healthcare architectures are particularly useful when multiple patients and devices must be managed centrally. Rather than storing all observations on a local workstation, measurements can be transmitted through a gateway to cloud services where they are validated, persisted, analyzed, and presented through dashboards. This model can simplify centralized administration, although it also creates dependencies on connectivity, cloud configuration, and information-security controls.

Deep learning has become an important approach for physiological signal analysis. CNNs can learn hierarchical representations from structured signals, while recurrent neural networks can model temporal sequences. LSTM networks were introduced to address difficulties in learning long-term dependencies in recurrent networks and have subsequently been applied to sequential data (Hochreiter and Schmidhuber, 1997) CNNs have also become a general-purpose deep-learning approach for learning representations directly from data (LeCun, *et al*. 2015)

For ECG analysis, PhysioNet provides multiple openly accessible physiological databases. The MIT-BIH Arrhythmia Database contains annotated two-channel ambulatory ECG recordings, and the MIT-BIH Long-Term ECG Database contains long-duration recordings with manually reviewed beat annotations (Moody and Mark, 2001) These resources demonstrate the availability of research data for future validation, but their use does not automatically make a proposed healthcare platform clinically validated.

Electronic health-record and critical-care datasets are also available through PhysioNet, including MIMIC and eICU resources under their respective access conditions Such datasets can support future experiments involving broader clinical variables, but access requirements, preprocessing decisions, cohort definitions, and ethical considerations must be documented.

The proposed work differs from a pure signal-classification study because the central research problem is an end-to-end cloud monitoring architecture. The deep-learning model is treated as one component within a larger pipeline consisting of sensing, ingestion, preprocessing, prediction, alert prioritization, visualization, and secure data management.

Table 1: Functional Comparison of Healthcare Monitoring Approaches

Approach	Data handling	Prediction	Scalability	Main limitation
Manual/periodic monitoring	Local/manual records	Human interpretation	Low	Delayed updates and workload
Standalone wearable	Device/local app	Limited or rule-based	Medium	Restricted central coordination
Cloud monitoring	Centralized cloud	Rules/ML possible	High	Connectivity and security dependency
Cloud + deep learning	Cloud + model pipeline	Learned risk patterns	High	Requires validated data and model governance

Table I summarizes the architectural distinction rather than reporting clinical accuracy.

The purpose is to show why deep learning should be integrated with, rather than considered a substitute for, the monitoring infrastructure.

Challenges in Cloud-Based Healthcare Monitoring

A dependable healthcare monitoring platform must address technical, clinical, and organizational challenges. The following issues are central to the proposed design.

A. Data Quality and Sensor Reliability

Physiological sensors may produce missing values, motion artifacts, disconnected intervals, saturation, or implausible measurements. A deep-learning model trained on unvalidated observations may learn artifacts instead of clinically meaningful patterns. The proposed pipeline therefore places validation and preprocessing before model inference.

Quality flags should be retained so that the system can distinguish a genuinely abnormal observation from an unreliable measurement.

B. Temporal Nature of Physiological Data

Health risk is rarely represented by a single measurement alone. A heart-rate value may be normal for one patient and concerning for another depending on recent history, activity, medications, and other variables. Consequently, the proposed model operates on time windows rather than isolated records. Sequential modeling is intended to capture changes and persistence over time.

C. False Alarms and Alert Fatigue

A monitoring system that generates excessive alerts can overwhelm users and reduce trust. Conversely, suppressing alerts too aggressively can delay attention to potentially important changes. The proposed framework separates model probability from alert priority. A probability threshold, persistence rule, signal-quality check, and escalation policy can be combined before an alert is delivered.

D. Privacy and Security

Healthcare information is sensitive, and cloud deployment increases the number of components that must be protected. Identity management, encryption in transit and at rest, least-privilege access, audit logging, retention policies, and secure device authentication should be considered during system design. The uploaded reference paper similarly identifies privacy of stored personal information as a practical concern in automated systems; in healthcare, the consequences of inadequate protection are substantially more serious.

E. Connectivity and Latency

Cloud processing requires a reliable communication path between the patient-side gateway and the cloud service. Network interruptions can create missing intervals or delayed alerts. A resilient implementation can buffer observations locally and synchronize them when connectivity returns. For time-critical scenarios, a hybrid edge-cloud architecture may perform basic safety checks locally while using the cloud for centralized analytics.

F. Explainability and Clinical Trust

A deep-learning model can produce a risk score without providing an intuitive clinical explanation. The proposed dashboard should therefore display the time window used for prediction, relevant physiological trends, data-quality indicators, and model confidence. Such information can help clinicians interpret the output, although model explanations should not be treated as proof of causation.

G. Interoperability and Deployment

A research prototype may use a simple database and dashboard, but a production system must integrate with existing healthcare workflows. Standardized data representations, consistent timestamps, patient identifiers, device metadata, and auditable interfaces are necessary for reliable exchange. The proposed architecture intentionally separates acquisition, storage, analytics, and presentation so individual components can later be replaced or integrated with healthcare standards.

Proposed Cloud-Based Healthcare Monitoring System

The proposed system consists of five major layers: patient-side data acquisition, communication and gateway services, cloud ingestion and storage, deep-learning analytics, and user-facing alert/dashboard services. The design is modular so that additional physiological parameters or alternative deep-learning models can be introduced without redesigning the entire system.

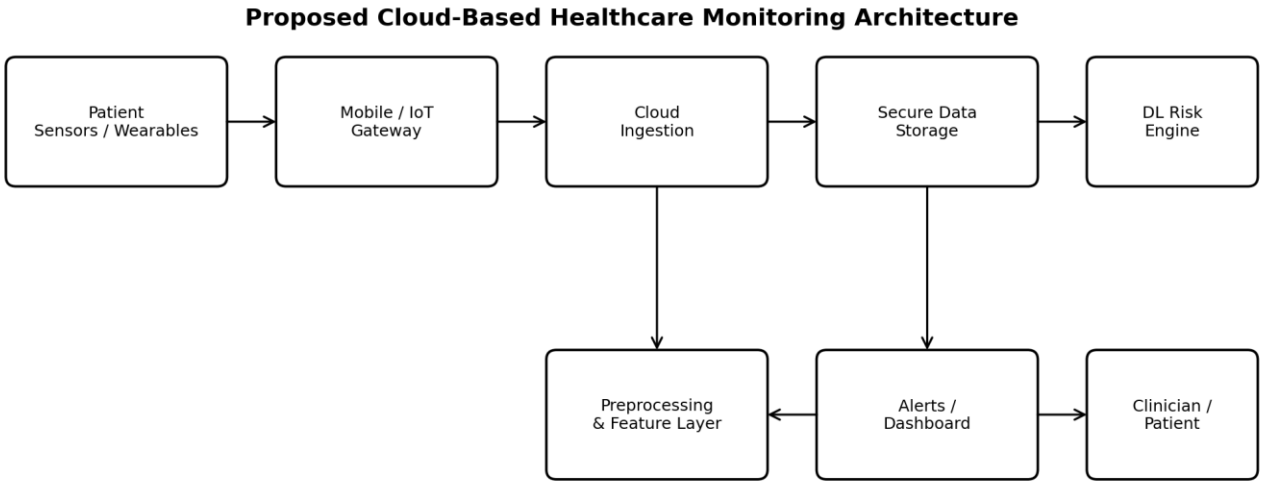

Fig. 1: Proposed architecture of the cloud-based healthcare monitoring system

A. Patient Data Acquisition Layer

The acquisition layer collects physiological observations from wearable sensors, bedside devices, mobile applications, or manually entered measurements. Candidate parameters include heart rate, SpO_2, body temperature, blood pressure, respiratory rate, and ECG-derived measurements. Each observation should include a timestamp, patient identifier, device identifier, measurement value, and quality metadata where available.

The system should avoid assuming that every patient has the same set of sensors. A modular data model can represent missing channels explicitly rather than silently replacing them with zeros. This is important because missingness itself may reflect device or connectivity problems.

B. Communication and Gateway Layer

A mobile phone, edge computer, or IoT gateway can receive observations from local devices and forward them to the cloud. The gateway can perform basic validation, timestamp synchronization, buffering, and secure transmission. For intermittent connectivity, local buffering prevents immediate data loss.

C. Cloud Ingestion and Storage Layer

The cloud layer receives measurements through an authenticated application interface or message broker. Incoming observations are validated and stored in a patient-centered data repository. A separate event stream can be used for real-time inference, while a persistent database maintains historical records for trend analysis and model development.

The cloud architecture should separate raw data from derived data. Raw observations should be retained according to an appropriate policy, while processed features, predictions, alert events, and audit records should be stored with provenance information.

D. Deep-Learning Analytics Layer

The proposed analytical engine uses a CNN-LSTM architecture. The CNN component extracts local patterns from a normalized time window, while the LSTM component processes the resulting sequence to learn temporal dependencies. The final dense layer produces a risk probability or a multi-class risk score depending on the target task.

E. Dashboard and Alert Layer

The dashboard presents patient status, recent trends, data-quality warnings, model output, and alert history. Alerts can be prioritized into informational, moderate-risk, and high-priority categories. The system should clearly label predictions as decision-support outputs and provide escalation pathways to authorized healthcare professionals.

Methodology

The proposed methodology follows a reproducible pipeline from data acquisition to alert generation. The workflow is intentionally defined so that experimental results can be added later without changing the conceptual architecture.

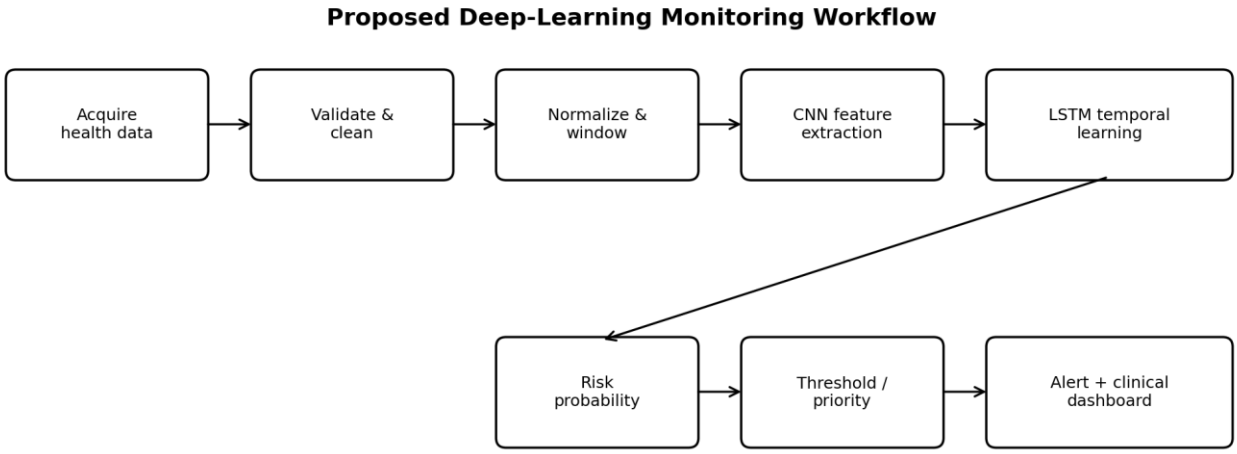

Fig. 2: Proposed deep-learning methodology and alert-generation workflow

A. Data Representation

Let the multivariate physiological observation at time t be represented as x_t = [x_t1, x_t2, ..., x_td], where d denotes the number of monitored channels. A fixed-length window containing T observations is represented as X = [x_1, x_2, ..., x_T]. The target y may represent a binary risk state, such as lower-risk versus elevated-risk, or a multi-class state such as normal, moderate, and high priority.

The framework does not prescribe a specific clinical label without a validated dataset. The target definition must be determined from the selected dataset and research question. This prevents the paper from making an unsupported clinical diagnosis claim.

B. Preprocessing

Preprocessing consists of five main operations: timestamp validation, missing-value handling, outlier checking, normalization, and temporal windowing. Continuous channels can be normalized using training-set statistics. For a feature x, standardization may be written as:

$$z = (x - \mu\_train) / \sigma\_train$$

where μ_train and σ_train are computed only from the training partition. This prevents information from the test set from leaking into model training.

Missing observations should be handled according to the semantics of the channel. Short gaps may be interpolated when scientifically justified, while long gaps can be flagged as unavailable. The model input should retain a quality indicator whenever possible.

C. CNN Feature Extraction

For one-dimensional physiological sequences, a convolutional layer can apply learnable filters over neighboring observations. For input sequence X, a simplified convolution operation can be expressed as:

$$h_i = \varphi(\Sigma_k w_k x_{(i+k)} + b)$$

where w represents a learned kernel, b is a bias term, and φ is an activation function. Multiple filters can learn different local waveform or trend patterns.

D. LSTM Temporal Modeling

The convolutional feature sequence is supplied to an LSTM layer. The LSTM maintains an internal state and uses gating mechanisms to control the flow of information through time. In general form, the recurrent computation can be represented as $h_t = LSTM(h_{t-1}, c_{t-1}, z_t)$, where z_t is the CNN-derived feature vector. The LSTM output is then passed to one or more dense layers.

A compact model can contain one or two convolutional blocks followed by an LSTM layer, dropout for regularization, and a final sigmoid or softmax layer depending on the target definition.

E. Training Strategy

The dataset should be divided at the patient level whenever repeated measurements from the same individual are present. This reduces the risk that highly similar observations from one patient appear in both training and test sets. Class imbalance should be measured before training. Weighted loss, balanced sampling, or threshold optimization can be considered when justified by the dataset.

Training should include a validation partition for hyperparameter selection and an independent test partition for final evaluation. Early stopping can be used to reduce overfitting. All preprocessing parameters must be learned from the training data only.

Model Design, Risk Scoring and Cloud Deployment

A. Proposed CNN-LSTM Model

A representative architecture is shown below. The exact number of filters, kernel sizes, recurrent units, dropout rates, and learning rate should be selected experimentally rather than presented as universally optimal.

Table 2: Representative CNN-LSTM configuration for experimental implementation

Layer	Example configuration	Purpose
Input	T × d physiological window	Receives synchronized measurements
Conv1D	32 filters, small kernel	Local feature extraction
Activation	ReLU	Nonlinear representation
Pooling	Max/average pooling	Dimension reduction
Conv1D	64 filters, small kernel	Higher-level local patterns
LSTM	64 units	Temporal dependency learning
Dropout	0.2–0.5 candidate range	Regularization
Dense	32 units	Feature integration
Output	Sigmoid/Softmax	Risk probability/class

.

B. Risk Score

For binary classification, the final layer can produce a probability $p \in [0,1]$. A simple decision rule is:

$$Risk = 1, \text{ if } p \geq \tau; \text{ otherwise } Risk = 0$$

where τ is a threshold selected using validation data and the intended operating point. The threshold should not be chosen only to maximize accuracy when false negatives and false positives have different consequences.

For a multi-class model, the softmax function can produce class probabilities. The dashboard can display both the highest-probability class and the probability distribution so that uncertainty is visible.

C. Alert Prioritization

A practical monitoring system should combine model output with data quality and persistence. One possible conceptual score is:

$$Priority = f(\text{model probability, persistence, signal quality, patient context})$$

This is deliberately presented as a design principle rather than a validated clinical formula. A research implementation can define and evaluate a concrete alert policy using an appropriate labeled dataset and domain expertise.

D. Cloud Deployment

The model can be deployed as a cloud inference service behind an authenticated API. The service receives a validated window, performs the same preprocessing used during training, returns a prediction, and records the model version and timestamp. Model versioning is essential because predictions generated by different model versions should remain distinguishable.

The cloud database can store patient profiles, observations, quality flags, predictions, alert events, and audit logs. A message queue or event-driven component can separate data ingestion from model inference, allowing the system to remain responsive when multiple patients transmit data simultaneously.

A dashboard can retrieve the latest status through an application programming interface. Role-based access can restrict clinicians to authorized patients, while administrative users can manage device registration and system configuration. The design should also support audit trails for data access and alert acknowledgement.

E. Security Design

Security should be considered across the complete data path: device authentication, encrypted communication, secure API endpoints, encrypted cloud storage, identity management, least-privilege access, logging, backup, and retention. Sensitive identifiers should be minimized in machine-learning datasets where possible. Research datasets should be de-identified or accessed under the terms established by their providers.

Evaluation Framework and Comparative Analysis

Because this paper presents a proposed framework without a completed clinical dataset experiment, the evaluation section defines the measurements that should be used in future implementation. No numerical model performance is claimed.

A. Classification Metrics

For binary risk classification, the confusion matrix provides true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Accuracy, precision, recall, and F1-score can be computed as follows:

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

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

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

$$F1 = 2 \times Precision \times Recall / (Precision + Recall)$$

For healthcare risk screening, recall/sensitivity can be particularly important when missing a high-risk case is more consequential than generating an additional review alert. Specificity, negative predictive value, calibration, and ROC-AUC or PR-AUC should also be considered where appropriate.

B. System-Level Metrics

Model quality alone does not describe a cloud monitoring platform. System evaluation should include end-to-end latency, data ingestion rate, missing-data rate, alert delivery delay, cloud resource utilization, uptime, and recovery after connectivity interruption. The evaluation should also measure the number of false alerts per patient-day when continuous monitoring is used.

Table 3: Proposed evaluation matrix

Dimension	Metric	Purpose
Prediction	Accuracy / F1	Overall classification performance
Safety-oriented	Recall / sensitivity	Measure missed positive cases
Specificity	Specificity	Measure false-positive tendency
Calibration	Brier score / calibration	Assess probability reliability
Latency	End-to-end delay	Assess timeliness of alerts
Reliability	Data loss / uptime	Assess operational stability
Scalability	Throughput / concurrent users	Assess cloud capacity
Usability	Alert acknowledgement / feedback	Assess practical workflow

C. Comparison of Monitoring Approaches

Feature	Manual monitoring	Cloud monitoring	Proposed cloud + CNN-LSTM
Continuous collection	Limited	Yes	Yes
Centralized storage	No/limited	Yes	Yes
Temporal pattern learning	Human review	Optional	Yes
Automated risk score	No	Rule/ML dependent	Deep learning
Remote dashboard	Limited	Yes	Yes
Scalability	Low	High	High
Clinical validation needed	Workflow-based	Yes	Yes

The comparison is architectural and does not imply that the proposed method is clinically superior. Clinical superiority would require a prospective or appropriately designed retrospective evaluation against a meaningful baseline.

D. Future Experimental Plan

A future study can begin with a public physiological dataset and then reproduce the complete pipeline from patient-level splitting through independent testing. For ECG-focused experiments, PhysioNet's MIT-BIH resources provide established research data; the MIT-BIH Long-Term ECG Database, for example, contains seven long-duration ECG recordings with manually reviewed beat annotations. The experimental protocol should report dataset characteristics, preprocessing, class definitions, patient split strategy, hyperparameters, baseline models, confidence intervals where appropriate, and error analysis.

Discussion

The proposed architecture combines three ideas that are often treated separately: remote physiological data acquisition, cloud-based healthcare information management, and deep-learning-based temporal analysis. Integrating these components creates an end-to-end research problem rather than a standalone classification task.

The main advantage of the cloud layer is centralized coordination. A monitoring service can collect observations from multiple patients, maintain historical records, run inference services, and expose a unified dashboard. This can support scalable research and remote workflows. However, centralized infrastructure also increases the importance of security, availability, and governance.

The CNN-LSTM design is motivated by the structure of physiological time series. Local patterns may occur within short signal segments, while clinically relevant changes may develop over longer intervals. A convolutional front end can transform the raw sequence into a feature representation before recurrent processing. Nevertheless, CNN-LSTM should be treated as a hypothesis to evaluate, not as a guaranteed best architecture. Temporal convolutional networks, GRUs, transformers, classical machine-learning baselines, and simpler statistical methods should be included in a serious comparative experiment.

Another important issue is the difference between statistical prediction and clinical usefulness. A model can obtain a high test score on a dataset while failing under device changes, population shifts, missing channels, or different clinical settings. Therefore, future work should evaluate robustness across patient groups, devices, environments, and time periods. External validation is especially important before any claim of general clinical utility.

Explainability should also be addressed. A dashboard that simply shows 'high risk' may not provide enough information for a clinician to assess the output. Future implementations can expose recent trends, contributing features, signal-quality indicators, and model uncertainty. Such explanations should be evaluated for usefulness rather than assumed to be correct simply because they are generated by an algorithm.

Privacy is another major design consideration. The uploaded reference paper identifies the storage of personal information and photographs as a system-level privacy concern. In healthcare, patient observations are substantially more sensitive, so privacy should be designed into the system from the beginning rather than added after implementation. Data minimization, access control, retention policies, auditability, and secure transmission should therefore be treated as architectural requirements.

The absence of experimental results in this paper is intentional. Reporting fabricated accuracy values would make the paper appear complete while weakening its scientific validity. Instead, the work defines a reproducible methodology and an evaluation plan. Once an approved dataset is selected, the numerical results can be inserted into the comparative tables and discussion without changing the core architecture.

Limitations

First, the proposed framework is not a clinically validated diagnostic system. Second, no patient dataset was analyzed in the present manuscript, so performance values cannot be reported. Third, the proposed risk labels are generic and must be operationalized using a clinically meaningful target definition. Fourth, cloud dependence may introduce latency or service-availability issues. Fifth, deep-learning models can be sensitive to distribution shifts, sensor differences, missing data, and class imbalance. Finally, successful deployment requires appropriate clinical, ethical, security, and regulatory processes beyond the software architecture described here.

Future Scope

Future work can proceed in several directions. A first step is to validate the architecture using an openly accessible physiological dataset and compare CNN-LSTM against CNN, LSTM, GRU, transformer, and classical machine-learning baselines. A second step is multimodal fusion, combining ECG or other signals with vital signs and contextual variables. A third direction is edge-cloud collaboration, where urgent lightweight checks run

locally and deeper analysis runs in the cloud. Federated learning can also be investigated when data cannot be centrally pooled. Additional work should address explainable AI, model calibration, uncertainty estimation, privacy-preserving learning, continuous model monitoring, and prospective clinical evaluation.

Conclusion

This paper presented a proposed cloud-based healthcare monitoring system using a CNN-LSTM deep-learning architecture for early detection of patient health risks. The framework combines physiological data acquisition, secure gateway communication, cloud ingestion and storage, preprocessing, temporal deep-learning analysis, risk scoring, and clinician-facing alerts. The architecture is designed to support scalable remote monitoring while keeping the analytical component modular.

The proposed methodology emphasizes that healthcare AI should not be evaluated only by model accuracy. Data quality, patient-level separation, false alarms, latency, security, privacy, explainability, and external validation are equally important. The paper therefore avoids unsupported numerical claims and instead provides a complete experimental plan that can be implemented using an appropriate public dataset or an ethically approved clinical dataset.

The resulting system is best viewed as a decision-support framework rather than an autonomous diagnostic tool. With future validation, robust security controls, clinically meaningful labels, and external testing, the architecture could serve as a foundation for research into intelligent remote patient monitoring and cloud-enabled digital healthcare.

References

World Health Organization Regional Office for Europe, “Scaling up telemedicine in the WHO European Region,” Policy Brief, 2025.

R. S. H. Istepanian, S. Hu, N. Philip, and M. Sungoor, “The potential of Internet of Things (IoT) for healthcare,” in Proc. 2011 5th Int. Conf. on Pervasive Computing Technologies for Healthcare, 2011.

A. Islam, M. M. Rahman, M. A. Rahman, M. A. Hossain, and M. M. Hossain, “Internet of Things for Health Care: A Comprehensive Survey,” IEEE Access, vol. 3, pp. 678–708, 2015.

S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, pp. 436–444, 2015.

G. B. Moody and R. G. Mark, “The impact of the MIT-BIH Arrhythmia Database,” IEEE Engineering in Medicine and Biology Magazine, vol. 20, no. 3, pp. 45–50, 2001.

PhysioNet, “MIT-BIH Long-Term ECG Database, version 1.0.0,” PhysioNet, DOI: 10.13026/C2KS3F.

PhysioNet, “PhysioNet Databases,” MIT Laboratory for Computational Physiology, accessed 2026.

J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013.

V. M. R. Penmetsa and S. S. K. Venkatesh, “Cloud computing in healthcare: A review,” International Journal of Computer Applications, vol. 116, no. 1, 2015.

V. Rajkomar, J. Dean, and I. Kohane, “Machine learning in medicine,” New England Journal of Medicine, vol. 380, pp. 1347–1358, 2019.

Citation:

S.Brindha Devi. (2026). Cloud-Based Healthcare Monitoring System using CNN-LSTM for Early Detection of Patient Health Risks International Journal of Current Science Research (IJCSR) e-ISSN: 2454-5422, 12(7), 10-25. Published by Dr. BGR Publications.