Traitement du Signal
Vol. 42, No. 6, December, 2025, pp. 3601-3614
Journal homepage: http://iieta.org/journals/ts

An Optimized Model for Cardiovascular Risk Prediction in Wearable Devices

Bisna N. Divakaran[1*], Sona Pulikkodan[2], Ajay James[3], Dileesh E. Dharmajan[1]

[1] Department of Computer Science and Engineering, Government Engineering College Thrissur, APJ Abdul Kalam Kerala Technological University, Thiruvananthapuram 695016, India
[2] Department of Computer Science and Engineering, Vimal Jyothi Engineering College, Kannur 670632, India
[3] Department of Computer Science and Engineering, Government Engineering College, Idukki 685603, India

Corresponding Author Email: bisna@gectcr.ac.in

https://doi.org/10.18280/ts.420642

Received: 20 July 2025
Revised: 12 November 2025
Accepted: 1 December 2025
Available online: 31 December 2025

Keywords:
heart disease prediction, wearable devices, machine learning, deep learning, TinyML, pruning, quantization, CNN-LSTM, IoT

ABSTRACT

Cardiovascular disease remains one of the foremost causes of mortality worldwide, emphasizing the urgent need for accurate and energy-efficient early risk prediction methods. The growing availability of wearable devices capable of continuously capturing physiological data, such as heart rate, blood pressure, sleep duration, and activity levels, presents a powerful opportunity for proactive health monitoring. In this study, we propose a lightweight, intelligent system for predicting heart disease risk by leveraging hybrid Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) architecture. The model is specifically optimized for edge deployment using TinyML techniques, including pruning and quantization, to reduce computational complexity while maintaining high predictive performance. This research focuses on centralized training using preprocessed multimodal wearable data, addressing challenges such as data imbalance and real-time resource constraints. The proposed system achieves high predictive performance while significantly improving efficiency for wearable deployment. The baseline CNN-LSTM model attains 95.27% accuracy, with the pruned and quantized versions maintaining 95.23% and 95.14%, respectively. Model size is reduced from 610MB to 190MB, power consumption drops from 16.74W to 7.90W, and inference time improves from 71 s to 48 s, demonstrating that the optimized model supports real-time, low-power cardiovascular-risk prediction on edge devices.

1. INTRODUCTION

Cardiovascular diseases (CVDs) remain a major global health challenge, causing nearly 18 million deaths each year according to the World Health Organization (WHO). They contribute significantly to morbidity and mortality worldwide [1]. Aging populations and changing lifestyles continue to worsen this burden. The impact of CVDs extends beyond individual patients, affecting families and healthcare systems as well. This widespread prevalence highlights the urgent need for effective, timely, and accurate detection methods. Early intervention can help save lives and reduce the strain on healthcare resources.

One of the critical challenges in CVD management is the limited availability of continuous cardiovascular monitoring, particularly for high-risk individuals who remain asymptomatic. Traditional diagnostic methods, such as periodic check-ups and tests, often fail to detect issues early enough for timely intervention. Given the dynamic nature of cardiovascular health, it is imperative to adopt solutions that offer real-time monitoring, enabling continuous assessment of a patient's health status. This would not only allow for the early detection of abnormalities but also provide valuable data for personalized treatment and prevention strategies.

The integration of wearable sensors and artificial intelligence (AI) has emerged as a promising approach to meet the need for continuous, real-time health monitoring. Wearable devices, such as fitness trackers, smart watches, and other biosensors, are capable of gathering vital signs such as heart rate, blood pressure, physical activity levels, and sleep patterns. These sensors provide a continuous stream of data, which, when analyzed, can reveal patterns indicative of cardiovascular risk. However, despite these advancements, there are significant challenges in utilizing deep learning models on wearable devices, primarily due to the limited processing power and energy constraints of these devices.

Traditional deep learning models, particularly those that process sequential and spatial data, require substantial computational resources, which are often beyond the capabilities of typical wearable devices. Furthermore, processing these complex models in real-time consumes considerable energy, which can rapidly drain the device's battery, making continuous monitoring impractical.

To address these challenges, we present an approach that integrates Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks for predicting the risk of heart disease. CNNs are particularly effective in extracting meaningful features from data with spatial patterns, such as

electrocardiogram (ECG) signals or physical activity data. On the other hand, LSTMs are well-suited for modeling sequential dependencies in time-series data, such as heart rate and blood pressure trends. By leveraging the strengths of both architectures, the proposed hybrid CNN-LSTM model offers enhanced accuracy in forecasting cardiovascular risk.

To enable the deployment of the proposed hybrid model on wearable devices, we incorporate Tiny Machine Learning (TinyML) strategies with a focus on pruning and quantization techniques. Pruning enhances model efficiency by removing redundant or low-impact weights, thereby reducing model complexity. Quantization, meanwhile, lowers the bit-width of weights and activations, which minimizes memory usage and computational load. These optimizations collectively ensure that the deep learning model can run efficiently on devices with limited resources, enabling real-time and energy-efficient cardiovascular monitoring.

The objective of this work is to design a predictive system that delivers performance comparable to traditional healthcare solutions, while remaining computationally lightweight for seamless deployment on wearable devices. By addressing the power and memory limitations inherent to such platforms, this model paves the way for scalable and personalized cardiovascular monitoring in everyday settings.

This study offers several key contributions. First, it demonstrates how a hybrid CNN-LSTM architecture can be effectively adapted and optimized for deployment in real wearable-edge environments. This is validated through implementation and testing on a Raspberry Pi 4 Model B edge device.Secondly, it integrates TinyML techniques, which are structured pruning and 8-bit quantization to substantially reduce computational load and memory usage, enabling real-time inference under strict resource constraints. Thirdly, beyond evaluation on the primary wearable Kaggle dataset, the model's generalizability was further confirmed using the Medical Information Mart for Intensive Care IV (MIMIC-IV) Waveform dataset from PhysioNet [2], where 5-fold cross-validation established the robustness and stability of the optimized model.

Together, these advancements lay the foundation for a scalable, energy-efficient, and clinically reliable framework for continuous cardiovascular-risk monitoring.

2. LITERATURE SURVEY

Cardiovascular disease prediction has evolved significantly with advances in wearable sensing, Internet of Things (IoT) systems, and machine-learning methodologies. Early research focused on traditional statistical and machine-learning models applied to structured clinical datasets. Recent work has leveraged deep-learning architectures capable of extracting complex temporal and physiological patterns from continuous sensor data. Parallel developments in TinyML and edge computing have further enabled resource-efficient deployment of intelligent health-monitoring systems on wearable and low-power devices. This literature review synthesizes prior contributions across these domains, highlighting key methodologies and limitations.

Bhatt et al. [3] highlighted the continued relevance of classical machine-learning methods such as logistic regression, decision trees, and Support Vector Machines (SVM) for heart-disease prediction due to their interpretability and suitability for structured clinical data. Their approach incorporated essential preprocessing steps, including feature selection and discretization (e.g., binning age, blood pressure, and cholesterol), which helped capture nonlinear relationships and improved classifier performance. They also employed k-modes clustering to identify latent patterns in categorical medical data, further enhancing predictive accuracy.

Traditional ML approaches have additionally been applied to wearable-sensor data. Siirtola et al. [4] and Martin-Gonzalez et al. [5] used Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) to predict cardiac-related events from sleep-sensor signals. While initial user-independent models suffered from variability in individual sleep patterns, personalized models significantly improved performance by tailoring decision boundaries to each user's physiological characteristics.

Wearable and IoT technologies are rapidly reshaping cardiovascular monitoring by enabling continuous, non-clinical data collection and analytics. Singhal and Cowie [6] reviewed how wearables support heart-failure management, highlighting their promise and current limitations such as data validity and clinical integration. Perez-Pozuelo et al. [7] demonstrated wearable devices could detect sleep outside the clinic, supporting scalable early anomaly detection.

Alday et al. [8] presented the 2020 PhysioNet challenge on 12-lead ECG classification, underlining the integration of sensor data into predictive workflows. Lin et al. [9] provided a detailed survey of wearable sensors and devices for real-time cardiovascular disease monitoring, covering hardware, signals — ECG, Photoplethysmography (PPG) — and deployment platforms. De Zambotti et al. [10] showed how a commercially available wristband could capture sleep and cardiac function in adolescents, confirming the feasibility of consumer wearables in cardiovascular research. Sarmah [11] reported an IoT-based deep-learning system for heart-disease prediction, combining device data with modified neural networks, but warned of system-level risks like latency and data quality.

Ali et al. [12] proposed an ensemble deep-learning and feature-fusion approach using wearable and IoT-derived variables for heart-disease prediction, offering strong accuracy yet raising issues of interpretability. Kundrick et al. [13] applied machine learning to wearable fitness tracker data to predict hospitalizations and cardiovascular events. This demonstrated improved risk stratification using continuous real-world physiological signals. However, the study was limited by device-specific data and cohort dependence, which might affect generalizability.

Deperlioglu et al. [14] applied an autoencoder–Deep Neural Network (DNN) in a secure Internet of Health Things (IoHT) framework for disease diagnosis, trading transparency for performance. Pakhomov et al. [15] integrated electronic medical records with sensor data for heart-failure identification, improving prediction depth but facing interoperability and privacy hurdles.

Recent research on deep-learning approaches for heart-disease prediction has demonstrated significant advances in representation learning, feature augmentation, and hybrid network design. García-Ordás et al. [16] showed that deep neural architectures combined with feature augmentation substantially improved prediction accuracy on structured clinical datasets. Alqurashi et al. [17] integrated Predator Crow Optimization with deep neural networks to automate hyperparameter tuning and feature selection, achieving strong performance but at the cost of increased computational complexity.

Mohammad and Al-Ahmadi [18] proposed a hybrid Wavelet Transform–Convolutional Neural Network (WT-CNN) model that extracted wavelet-based ECG features before applying convolutional layers, reporting accuracy levels near 97%. This work highlighted the value of signal-driven deep learning. Shankar et al. [19] demonstrated that even relatively simple CNN architectures could outperform classical machine-learning models on heart-disease datasets when trained with proper regularization.

Ram Kumar et al. [20] introduced a hybrid CNN–DNN architecture that combined convolutional feature extractors with dense layers for final classification, achieving improved predictive performance but also noted potential overfitting risks on limited datasets. Collectively, these studies confirmed that deep-learning models can capture nonlinear feature interactions more effectively than traditional methods, while also emphasizing ongoing challenges related to dataset size, model interpretability, and robust cross-dataset validation.

Subashini and Kanaka Raju [21] presented an IoT-based heart-disease diagnosis framework that integrated physiological data collected via smart sensors with a hybrid learning pipeline combining gradient boosting for tabular features and a deep convolutional neural network for improved classification accuracy; however, the study was limited by evaluation on a restricted dataset and the lack of external clinical validation, which might affect generalizability. Abutalip et al. [22] proposed a machine-learning–driven heart-disease detection system using data acquired from wearable devices, where conventional ML classifiers were employed to identify cardiovascular risk patterns, but the work was constrained by limited feature diversity and short-term wearable data, reducing robustness across populations. Al Reshan et al. [23] developed a robust heart-disease prediction approach using hybrid deep neural networks, including CNN, LSTM, and a combined CNN–LSTM architecture, achieving high predictive performance on multiple benchmark datasets. Nevertheless, the model's reliance on curated public datasets and increased computational complexity posed challenges for real-time deployment and clinical interpretability.

Xia et al. [24] proposed an intelligent cardiovascular disease diagnosis framework that integrated Ant Colony Optimization for feature selection with a deep-learning–enhanced neural network, further optimized using Bayesian hyperparameter tuning to improve classification accuracy. Although the approach achieved strong performance, the combined use of metaheuristic optimization and deep learning increased computational complexity and training cost, which might limit its suitability for real-time or resource-constrained healthcare applications.

To make these models practical for wearables, researchers have applied TinyML techniques. Arooj et al. [25] explored structured pruning to reduce model complexity, while Neri et al. [26] showed that 8-bit quantization significantly reduced memory use and power consumption without major accuracy loss. Qureshi and Krishnan [27] demonstrated that TinyML-compatible CNN–LSTM models could deliver accurate, real-time predictions on microcontrollers, balancing performance with energy efficiency.

Recent work by Sun et al. [28] introduced a TinyML methodology for continuous, cuff-less blood-pressure estimation using only PPG signals. Their approach shrunk conventional CNN architectures (AlexNet, LeNet, SqueezeNet, ResNet, MobileNet) via pruning and quantization, and deployed them on constrained edge platforms. Their evaluation used thousands of ICU patient records and showed performance comparable to server-based systems while meeting the Association for the Advancement of Medical Instrumentation and the British Hypertension Society (AAMI/BHS) standards. The memory footprint was reduced to <1 MB and inference latency to around 10 ms on a Cortex-M microcontroller, although generalization across ambulatory settings remained a concern. Their work demonstrated the viability of TinyML for cardiovascular monitoring but highlighted the trade-off between model size and inter-subject robustness.

Mahardika et al. [29] proposed a CNN–LSTM architecture trained on the MIMIC-III arterial-blood-pressure (ABP) and PPG dataset. Their optimized configuration (5 convolutional layers + 1 LSTM + 2 dense layers) achieved a mean absolute error (MAE) of 7.89 ± 3.79 mmHg for systolic BP and 5.34 ± 2.89 mmHg for diastolic BP — meeting the AAMI and BHS limits. Deployment considerations included window segmentation (4 s, 500 points) and Principal Component Analysis (PCA) feature reduction. While the accuracy was promising, the architecture was still too heavy (around 10 M parameters) for ultra-low-power microcontroller deployment without further compression or dedicated hardware acceleration. This work underscored the value of hybrid CNN–LSTM models for cardiovascular regression tasks but also emphasized the need for TinyML-aware optimization.

Arthi and Krishnaveni [30] proposed a fog-enabled TinyML with explainable-AI pipeline for healthcare decision support. Their system achieved an F1-score of 0.93 for abnormal-health-event detection while employing Modified Lempel-Ziv-Welch (mLZW) data compression and Lightweight Shapley Additive explanations (SHAP) on edge/fog nodes. The paper reported memory usage of around 800 kB and latency of 30 ms for anomaly inference. However, the study used general health-sensor features rather than specialized cardiovascular signals and lacked prospective wearable deployment. It provided a valuable proof-of-concept for combining compression, interpretability, and TinyML in health monitoring.

Elhanashi et al. [31] provided a comprehensive survey of TinyML in embedded and IoT-based healthcare applications, covering more than 150 papers up to 2024. The review reported that the median parameter count of deployed TinyML models in health was around 35 k parameters, average memory footprint 256 kB, and average latency 20 ms. It also cited key challenges: lack of standardized evaluation benchmarks (e.g., sensitivity, calibration), limited cross-device generalization, and weak clinical validation. Their conclusions emphasized that, although TinyML is technically feasible and deployment-ready, clinically validated TinyML systems for cardiovascular applications remain scarce. Our work builds on these advances by deploying a compressed CNN–LSTM model for multi-class cardiovascular risk prediction, reporting clinically relevant metrics (sensitivity, specificity, calibration) and validating on edge hardware under realistic constraints.

3. PROPOSED METHOD

This section presents the step-by-step methodology employed in developing an energy-efficient heart disease prediction system using wearable sensor data and a hybrid CNN-LSTM model. The approach integrates data acquisition, preprocessing, model training, and TinyML-based

optimization to enable deployment on low-power wearable devices.

3.1 Dataset

The dataset employed in this study is the Wearables Dataset, publicly accessible on Kaggle. It comprises comprehensive health and lifestyle data collected from various wearable devices, including smartwatches, fitness trackers, and clinical-grade biosensors. The dataset includes a rich set of physiological indicators such as electrocardiogram (ECG) and photoplethysmography (PPG) signals, heart rate (HR), blood pressure (BP), sleep quality scores, and levels of physical activity, along with biometric and demographic information. Table 1 lists the attributes in the dataset. In total, the dataset contains 10,000 instances, each with 28 attributes that combine real-time sensor data and personal metadata, including age, gender, and medical history.

Table 1. Dataset attributes

Sl. No	Attribute
1	User_ID
2	Age
3	Gender
4	Weight
5	Height
6	Medical_Condition
7	Medication
8	Smoker
9	Alcohol_Consumption
10	Sleep_Duration
11	Deep_Sleep
12	REM_Sleep
13	Wakeups
14	Heart_rate
15	Blood_Oxygen
16	ECG
17	Calories_Consumed
18	Stress_level
19	Mood
20	Body_Fat
21	Health_Scan_Anomaly_Flag

To strengthen the reliability and clinical relevance of the proposed model, an additional external validation was performed using the MIMIC-IV Waveform Database from PhysioNet. This dataset provides high-fidelity ECG and vital-sign waveforms collected from real ICU patients, enabling robust assessment of the model's generalizability. Together, the two datasets support both development on wearable-style input data and validation on clinically grounded physiological signals.

3.1.1 Data preprocessing

The preprocessing phase ensures that the input data is clean, structured, and ready for model training. This process includes several critical steps:

Incomplete entries in categorical fields such as alcohol consumption or existing medical conditions were addressed using mean/mode imputation or deletion, depending on the extent of missingness and its impact on class distribution. All continuous numerical features (e.g., heart rate, blood pressure) were scaled to a standard range using min-max normalization to ensure uniformity in data representation and avoid bias during training. Time-series data such as ECG and PPG signals were filtered to remove outliers and irregularities that could skew the learning process.

To improve the efficiency and interpretability of the model, PCA [32] was applied during the feature selection stage. PCA is a statistical technique that transforms the original high-dimensional dataset into a smaller set of linearly uncorrelated variables known as principal components, which capture the maximum variance present in the data. By analyzing the cumulative explained variance, the top components that retained over 95% of the total variance were selected. This dimensionality reduction helped eliminate redundant and less informative features while preserving the essential patterns within the wearable physiological signals.

After applying PCA, the most informative features contributing to heart disease prediction included:

- Heart Rate Variability (HRV)
- Systolic and Diastolic Blood Pressure
- Oxygen Saturation (SpO2)
- Respiratory Rate
- Sleep Quality Score
- Activity Level Index

The selected components were then used as input for the CNN-LSTM architecture, allowing the model to focus on learning the most relevant spatial-temporal patterns in the data.

The classification task is centered on a target variable called the Anomaly Flag, which categorizes each record into one of four heart disease risk levels: Normal, Low, Medium, or High. A key challenge associated with this dataset is the imbalance in class distribution, where Normal and Low Risk instances significantly outnumber the Medium and High Risk categories. To address this issue and ensure robust model generalization, we applied the Synthetic Minority Oversampling Technique (SMOTE) [33] during the model training phase. This publicly available dataset ensures reproducibility and transparency for future research and validation.

We used k=5 nearest neighbors, and minority classes were oversampled until class distribution was approximately uniform. This method generates synthetic samples for minority classes by interpolating between existing instances, thus ensuring a more balanced distribution of risk categories.

Wearable sensors have known limitations, particularly reduced accuracy under motion, motion artifacts in PPG, environmental and placement variability, and user noncompliance, which affect measurement reliability. We addressed these issues by implementing preprocessing steps (bandpass filtering, motion-artifact detection and rejection, baseline correction), signal-quality indices (SQI) to exclude low-quality segments, and data augmentation during training to improve robustness to motion.

To provide a clear clinical basis for the Normal, Low, Medium and High risk labels used in this study, we mapped wearable-derived physiological features to established clinical thresholds and combined them into an interpretable composite score. Specifically, blood-pressure thresholds follow the 2017 American College of Cardiology (ACC) and the American Heart Association (AHA) [34] classification (Normal: SBP <120 & DBP <80mmHg; Elevated: SBP 120-129 & DBP <80mmHg; Stage-1 hypertension: SBP 130-139 or DBP 80-89mmHg; Stage-2 hypertension: SBP ≥140 or DBP ≥90mmHg). Oxygen saturation (SpO_2) thresholds were interpreted according to standard clinical guidance (normal ≥95%; values 90-94% considered concerning; <90% consistent with hypoxemia).

Heart-rate variability (HRV) reductions were treated as a

contributory risk indicator given its established association with cardiovascular morbidity and mortality [35]. These clinical cutoffs were used as inputs to a conservative composite rule. When multiple clinical flags were present, such as elevated blood pressure combined with low SpO_2 or markedly reduced heart rate variability, the subject was assigned to a higher risk category. In contrast, a single or borderline deviation resulted in a lower-risk label. All clinical feature extraction from validation waveforms included explicit preprocessing steps (filtering, artifact rejection, beat detection) and signal-quality checks prior to scoring to minimize spurious assignments.

3.2 System architecture

Figure 1 illustrates the end-to-end workflow of the proposed framework. The process begins with dataset collection followed by comprehensive preprocessing, including data cleaning, noise removal, reshaping, and handling of missing values. By applying PCA, relevant features are selected, and the data is divided into training and testing sets. SMOTE was applied on training dataset to address class imbalance.

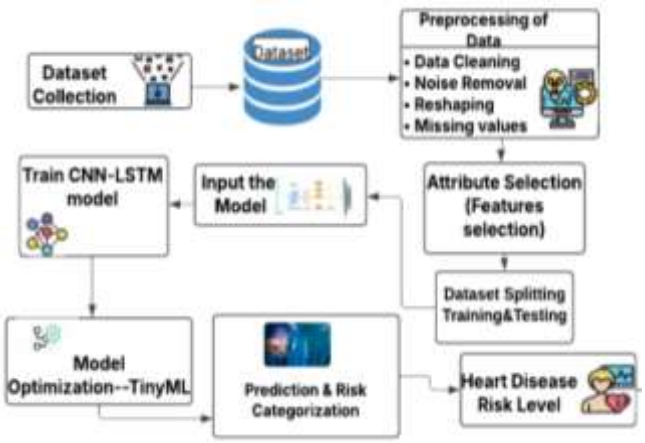


Figure 1. Proposed system architecture

A hybrid CNN-LSTM model is trained to learn both spatial and temporal patterns from wearable-sensor inputs, after which TinyML-based optimization techniques are applied to reduce model size and computational load.

As shown in Figure 2, the model begins with one-dimensional Convolutional layers (Conv1D), that analyze localized patterns in the input signals, such as the characteristic waveforms in ECG or PPG data. These layers help in extracting spatial features, which are crucial for recognizing anomalies or signal distortions indicative of cardiovascular issues.

The spatially filtered outputs are passed into Long Short-Term Memory (LSTM) layers, which are tailored to detect time-series trends and fluctuations over intervals. This is particularly useful in understanding how health metrics evolve, such as sudden spikes in heart rate or irregular heartbeat patterns. The model includes a fully connected (Dense) layer that fuses spatial and temporal insights. A final Softmax output layer categorizes each instance into one of the defined risk levels: Normal, Low, Medium, or High. The use of Softmax ensures that the model outputs a well-calibrated probability distribution across these classes.

As listed in Table 2, the model begins with an input layer structured as a 3D tensor of shape [B, T, F], where B is the batch size, T denotes the number of time steps, and F is the number of input features per time step (e.g., heart rate, ECG, PPG).

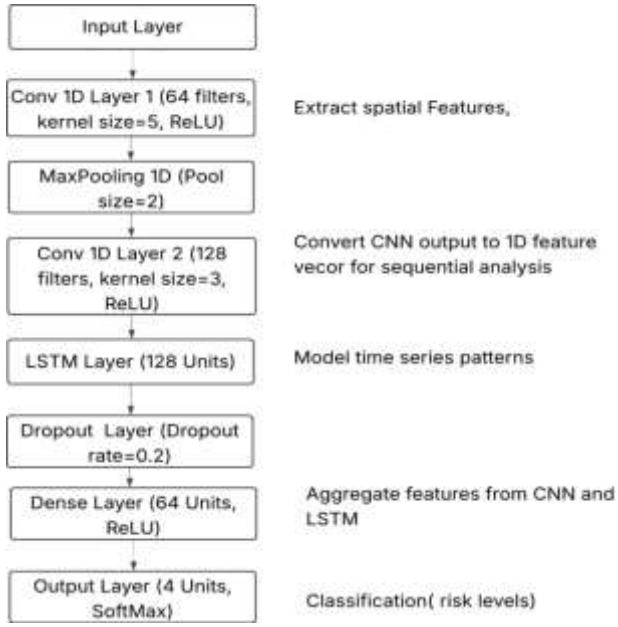


Figure 2. Proposed model architecture

Table 2. Model

Layer	Configuration / Parameters
Input Layer	[batch_size, time_steps, features]
Conv1D Layer 1	64 filters, kernel size=5, activation=ReLU
MaxPooling1D	Pool size=2
Conv1D Layer 2	128 filters, kernel size=3, activation=ReLU
LSTM Layer	128 units
Dropout Layer	Dropout rate=0.2
Dense Layer	64 units, activation=ReLU
Output Layer	4 units, activation=Softmax

The initial one-dimensional convolutional (Conv1D) layer utilizes 64 filters of size 5 with rectified linear unit (ReLU) activation to extract localized spatial features from the physiological signals.

The one-dimensional convolution operation is mathematically defined as follows:

$$y_t = \mathrm{ReLU}\left(\sum_{i=0}^{k-1} w_i \cdot x_{t+i} + b\right) \quad (1)$$

where,

x_{t+i} represents the input sequence segment starting at time $t + i$,

w_i denotes the convolutional kernel weights,

k is the kernel size (i.e., the number of input points considered at once),

b is the bias term,

ReLU(z)=max(0, z) is the Rectified Linear Unit activation function.

A MaxPooling1D layer is subsequently applied to reduce the dimensionality of the convolved feature map by selecting the maximum value within a sliding window of size 2. This process preserves salient features and decreases computational requirements.

The second Conv1D layer processes the feature map with 128 filters of size 3 and employs rectified linear unit (ReLU) activation. This configuration enables the model to capture

more complex spatial dependencies.

Next, the output from the convolutional block is passed to an LSTM (Long Short-Term Memory) layer with 128 units, designed to model the temporal dynamics in the physiological signals. The LSTM cell updates its internal state using the following equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$C'_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_c) \quad (4)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot C'_t \quad (5)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t \odot \tanh(C_t) \quad (7)$$

Here, f_t, i_t, and o_t are the forget, input, and output gates, C_t is the cell state, and h_t is the hidden state used in predictions.

This architecture allows the model to capture and retain significant temporal patterns in cardiac signals, including variability and abrupt changes. To enhance generalization and mitigate overfitting, a Dropout layer with a rate of 0.2 is incorporated. This layer randomly deactivates 20% of neurons during training, thereby reducing dependency on specific neural pathways. This is followed by a dense layer comprising 64 units with rectified linear unit (ReLU) activation, which consolidates the extracted spatial-temporal features into a unified representation. This representation is then forwarded to the final output layer, which employs a Softmax activation function to estimate the probabilities for four risk categories: Normal, Low, Medium, and High.

The Softmax function is defined as:

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \quad (8)$$

where, z_i is the logit (raw output) for class i, and C is the number of output classes. The class with the highest probability is selected as the final prediction.

3.3 Optimization

One of the major constraints in deploying deep learning models on wearable healthcare devices is the limitation in hardware capabilities such as processing power, memory availability, and energy efficiency.

To overcome these challenges, this study integrates Tiny Machine Learning (TinyML) techniques, specifically focusing on model pruning and quantization, as shown in Figure 3. This is to compress the model and optimize it for deployment on resource-constrained platforms without significantly compromising its predictive accuracy. Most wearable devices are designed to operate on low-power microcontrollers with limited battery life, making the direct implementation of computationally intensive deep neural networks impractical.

Model pruning is a technique aimed at reducing the size and complexity of a trained neural network by eliminating parameters that contribute minimally to the overall model performance. In this work, post-training weight pruning was employed to identify and remove low-magnitude weights within both the convolutional and recurrent layers. The rationale behind this method is that many neural network parameters have negligible influence on the model's output and can be removed without severely affecting its predictive capabilities. By setting these small-weight connections to zero and creating a sparse network representation, the model's size is substantially reduced.

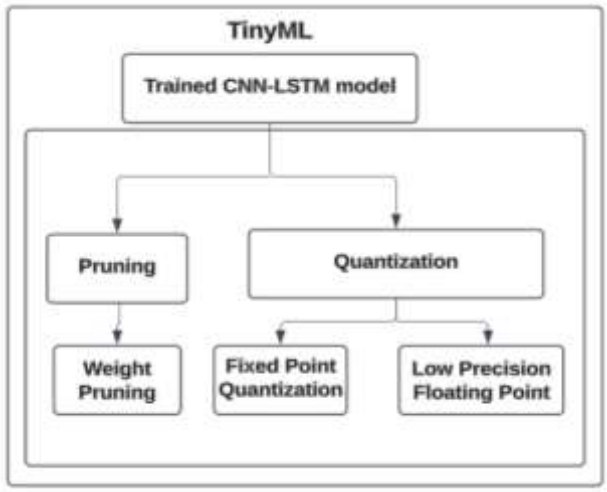


Figure 3. Optimization

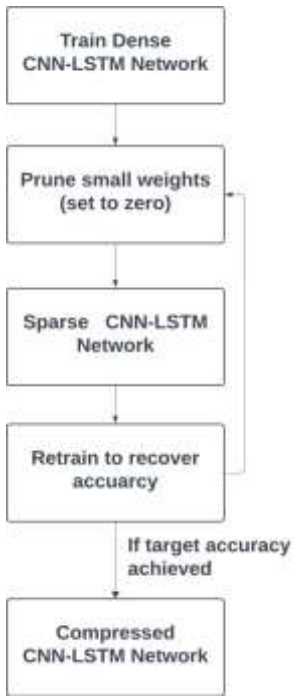


Figure 4. Pruning

Magnitude- based pruning is used here where weights are pruned when:

$$w_i = \begin{cases} 0 \text{ if } |w_i| < \tau \\ w_i \text{ Otherwise} \end{cases} \quad (9)$$

where, τ is the pruning threshold.

This creates a sparse model, improving efficiency while preserving important parameters. Sparsity S is defined as:

$$S = \frac{\#\{w_i = 0\}}{\#\{w_i\}} \quad (10)$$

As shown in Figure 4, following pruning, a fine-tuning process is conducted to restore any minor loss in accuracy, ensuring that the pruned model maintains robustness and generalization.

To further compress the pruned CNN-LSTM model, we employed k-means weight quantization as shown in Figure 5, which clusters the network weights into K representative centroids and replaces each weight with the index of its nearest centroid.

Formally, given the set of weights

$$w = \{w_i\}_{i=1}^{N} \tag{11}$$

Choose the number of quantization levels K (for 8-bit representation K=256).

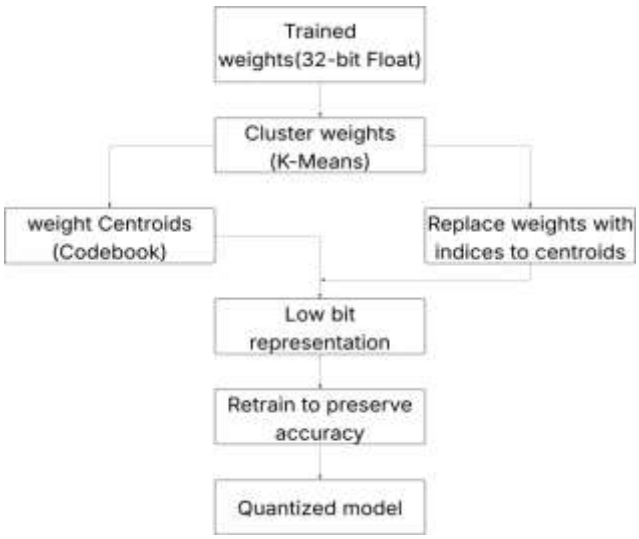


Figure 5. Quantization

The goal of k-means quantization is to find a codebook $C = \{c_1, c_2, c_3, c_4, c_5 \ldots c_K\} \subset R$ of K centroids and an assignment function k:{1,…,N}→{1,…,K} that minimizes the total squared reconstruction error:

$$\min J(C,k) \; where \; J(C,k) = \sum_{i=1}^{N} (w_i - c_{k(i)})^2 \tag{12}$$

Given, C the optimal assignment for each weight is the nearest centroid:

$$k(i) = \arg \min_{j \in \{1,\ldots K\}} |w_i - c_j| \tag{13}$$

And given assignments, centroids are updated by the sample mean of assigned weights:

$$c_j = \frac{1}{|S_j|} \sum_{i \in S_j} w_i, S_j = \{i : k(i) = j\}, \tag{14}$$

Iterating the assignment and centroid updates until convergence of J.

Quantized representation: each original weight w_i is replaced by an index k(i).

This transformation yields a model that is significantly smaller in size and faster in execution, with only a marginal loss in accuracy that remains within acceptable limits for clinical decision support systems.

4. EXPERIMENTS AND RESULTS

For fair comparison, the standalone CNN and LSTM baselines were implemented using standardized and lightweight architectures aligned with prior deep-learning studies on physiological signal analysis. The CNN model consisted of three 1D convolutional layers (Conv1D: 64 filters, kernel size 3; Conv1D: 128 filters, kernel size 3; Conv1D: 256 filters, kernel size 3), each followed by ReLU activation and max-pooling, and a final dense layer for classification.

The LSTM model comprised two stacked LSTM layers with 128 and 64 units, respectively, followed by a fully connected classification layer. Both models were trained using the same preprocessing pipeline, Adam optimizer, learning-rate schedule, batch size, and early-stopping strategy as the proposed CNN-LSTM. All models were evaluated using stratified 5-fold cross-validation, and performance metrics were reported as mean ± standard deviation across folds. The detailed results of this comparison are provided in Table 3 and illustrated in Figure 6.

Table 3. Performance metrics comparison with state-of-the art models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC
CNN (baseline)	92.6	90.8	91.9	91.3	0.941
LSTM (baseline)	93.1	91.2	92.4	91.8	0.946
Proposed CNN-LSTM	95.2	92.9	94.8	93.8	0.962

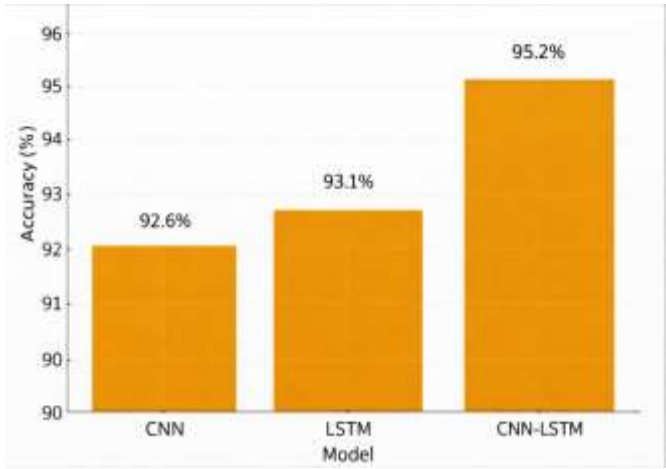


Figure 6. Performance metrics comparison with state-of-the art models

To examine the feasibility of deploying the optimized CNN-LSTM model in a real-world edge environment, we conducted performance testing on a Raspberry Pi 4 Model B (4GB RAM), a widely used embedded platform suitable for simulating wearable-device workloads. The Raspberry Pi 4 features a 1.5 GHz quad-core ARM Cortex-A72 CPU, LPDDR4 memory, and support for Python-based edge frameworks such as TensorFlow Lite, making it a suitable intermediate platform for validating lightweight machine-learning models before migration to ultra-low-power microcontroller units (MCUs).

The pruned and quantized model (190MB) was deployed on the Raspberry Pi using TensorFlow Lite, where we measured inference latency, memory consumption, and CPU usage during continuous streaming of wearable-like physiological data. The device operated within acceptable thermal and

power limits, and no throttling was observed during continuous execution, demonstrating that the optimized model is capable of real-time cardiovascular risk inference in a mobile-edge environment. Although the Raspberry Pi exceeds the resource constraints of commercial wrist-worn devices, it provides a realistic and controlled environment for evaluating embedded performance, identifying bottlenecks, and guiding further model compression or distillation for future deployment on low-power MCUs.

In addition to computational performance, we addressed key engineering and regulatory considerations essential for wearable medical systems. Since inference is performed entirely on-device, user physiological data does not need to be transmitted to cloud servers, reducing privacy and security risks.

The performance of the proposed CNN-LSTM model was assessed not only in terms of classification accuracy but also with respect to its suitability for deployment on resource-constrained wearable devices.

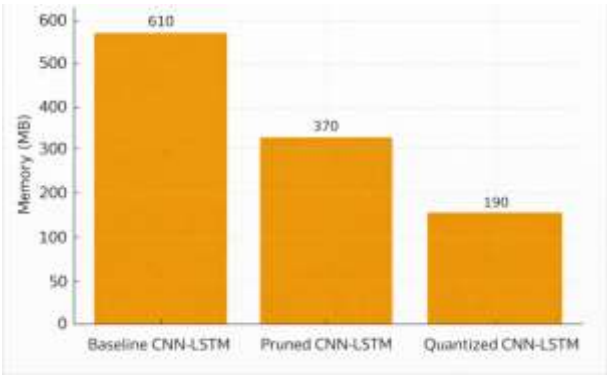


Figure 7. Memory usage

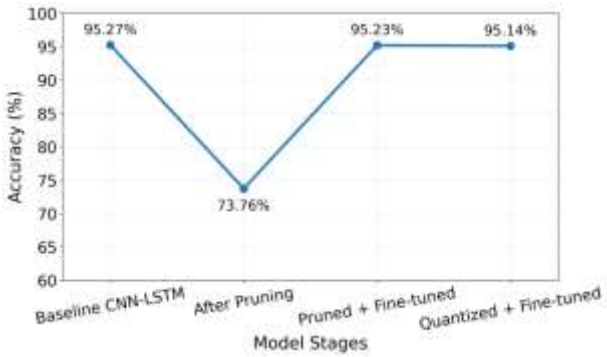


Figure 8. Model's accuracy

Figure 7 shows the memory usage across the baseline, pruned, and quantized versions of the model. The unoptimized baseline required approximately 610MB of memory, which is far beyond the capacity of typical edge devices. With pruning, memory consumption dropped to 370MB, and further quantization reduced it to just 190MB-representing a cumulative reduction of nearly 70%. This dramatic decrease in memory footprint demonstrates the effectiveness of TinyML techniques in preparing deep learning models for embedded environments where hardware limitations pose a significant concern.

Figure 8 highlights the classification accuracy of each model version. The baseline model achieved an accuracy of 95.27%, which decreased to 73.7% after pruning, then the fine-tuning process restored the accuracy back to 95.23% and the fine-tuning after quantization resulted in an accuracy of 95.14%. The results indicate that the core predictive capability of the CNN-LSTM architecture is largely preserved, even after aggressive model compression.

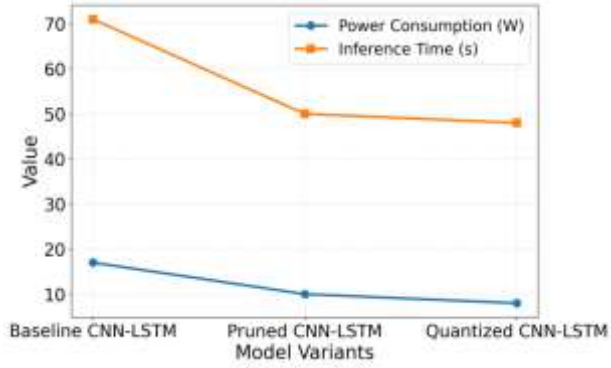


Figure 9. Power consumption and inference time

Figure 9 compares power consumption and inference time for the three configurations. The baseline model exhibited the highest power usage at 16.74 watts and the longest inference time at 71 seconds, which is impractical for continuous monitoring on battery-powered devices. The pruned model showed substantial improvement, reducing power draw to 10.2 watts and processing time to 50 seconds. The quantized version performed even better, consuming only 7.9 watts and completing inference in 48 seconds. These gains in speed and energy efficiency underscore the viability of TinyML-optimized deep learning for wearable deployment, enabling fast, on-device inference while conserving battery life.

Table 4. Performance metrics comparison (V1: Baseline CNN-LSTM, V2: Pruned CNN-LSTM (After fine-tuning), V3: Quantized CNN-LSTM (After fine-tuning))

Model Version	Accuracy (%)	Precision (%)	Sensitivity (%)	F1-Score (%)	Specificity (%)	AUC
V1	95.27	92.8	94.9	93.8	95.6	0.963
V2	95.23	93.2	93.5	93.3	95.3	0.958
V3	95.14	91.9	92.1	92.0	94.9	0.952

Table 5. Performance with statistically grounded metrics

Model	Metric	Mean	SD	95% CI
Baseline CNN-LSTM	Accuracy (%)	95.27	0.28	[94.81, 95.73]
	Precision (%)	92.8	0.35	[92.16, 93.44]
	Sensitivity (%)	94.9	0.31	[94.36, 95.44]
	F1-Score (%)	93.8	0.33	[93.26, 94.34]
	Specificity (%)	95.6	0.29	[95.13, 96.07]
	AUC	0.963	0.004	[0.956, 0.970]
Pruned CNN-LSTM	Accuracy (%)	95.23	0.32	[94.68, 95.78]
	Precision (%)	93.2	0.37	[92.53, 93.87]
	Sensitivity (%)	93.5	0.34	[92.96, 94.04]
	F1-Score (%)	93.3	0.35	[92.76, 93.84]
	Specificity (%)	95.3	0.31	[94.77, 95.83]
	AUC	0.958	0.005	[0.949, 0.967]
Quantized CNN-LSTM	Accuracy (%)	95.14	0.36	[94.57, 95.71]
	Precision (%)	91.9	0.39	[91.23, 92.57]
	Sensitivity (%)	92.1	0.37	[91.49, 92.71]
	F1-Score (%)	92.0	0.38	[91.39, 92.61]
	Specificity (%)	94.9	0.33	[94.36, 95.44]
	AUC	0.952	0.006	[0.941, 0.963]

Table 4 presents a comparison of clinically relevant

diagnostic metrics for the baseline, pruned, and quantized CNN-LSTM models. The baseline model demonstrates the strongest overall performance, with high sensitivity and specificity, indicating excellent ability to correctly identify both at-risk and normal individuals. The pruned model retains performance very close to the baseline, showing that model compression does not significantly affect its diagnostic reliability. The quantized model exhibits a slight reduction in sensitivity and specificity, but still maintains strong AUC values, confirming that it continues to provide clinically meaningful discrimination between risk levels.

Table 5 summarizes the performance of the three model variants with statistically grounded metrics. The baseline model achieved the highest accuracy (95.27%±0.28, 95% CI: 94.81-95.73) and AUC (0.963±0.004, CI: 0.956-0.970), while the pruned and quantized models showed only marginal reductions across precision, sensitivity, specificity, and AUC. The narrow confidence intervals and low standard deviations across all metrics indicate strong stability of the results and confirm that pruning and quantization introduced only minimal performance degradation. These statistical measures reinforce the reliability and robustness of the proposed TinyML-optimized model versions.

To ensure clinical relevance and dataset independence, the proposed model was further validated on the MIMIC-IV Waveform Database (PhysioNet), which contains clinically recorded ECG, PPG, arterial blood pressure (ABP), respiration, and SpO_2 signals.

All validation data underwent comprehensive preprocessing. This included noise filtering, baseline correction, and resampling. Following preprocessing, feature extraction was performed to compute heart rate, heart rate variability (HRV), blood pressure indices, oxygen saturation, and respiration rate. These features were consistent with the feature set used in the Kaggle Wearables dataset.

These features were normalized using the same scaling parameters to maintain uniformity across datasets. The model achieved an average accuracy of 93.8%±0.3**,** precision of 91.6%±0.3, recall of 93.1%±0.3, F1-score of 92.3%±0.3, and AUC of 0.953±0.002, confirming its ability to generalize effectively from wearable to clinical data. These results validate that the proposed CNN-LSTM model remains stable and accurate across heterogeneous signal sources when appropriate preprocessing and feature alignment are applied.

Table 6. Cross-validation results of the proposed CNN-LSTM model on MIMIC-IV dataset

Fold	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC
Fold 1	93.4	91.2	92.6	91.9	0.950
Fold 2	93.8	91.5	93.0	92.2	0.953
Fold 3	94.1	91.9	93.4	92.6	0.956
Fold 4	93.7	91.4	93.1	92.2	0.952
Fold 5	93.9	91.8	93.3	92.5	0.954
Mean±SD	93.8±0.3	91.6±0.3	93.1±0.3	92.3±0.3	0.953±0.002

Table 6 explores the impact of varying learning rates on the quantized model's accuracy, precision, recall, and F1-score. The five-fold cross-validation results demonstrate stable and consistent model performance across all evaluation metrics. The model achieved a mean accuracy of 93.8% ± 0.3%, with precision, recall, and F1-score of 91.6% ± 0.3%, 93.1% ± 0.3%, and 92.3% ± 0.3%, respectively, indicating a balanced classification performance. Additionally, the high mean AUC of 0.953 ± 0.002 reflects strong discriminative capability and robustness of the model across different validation folds.

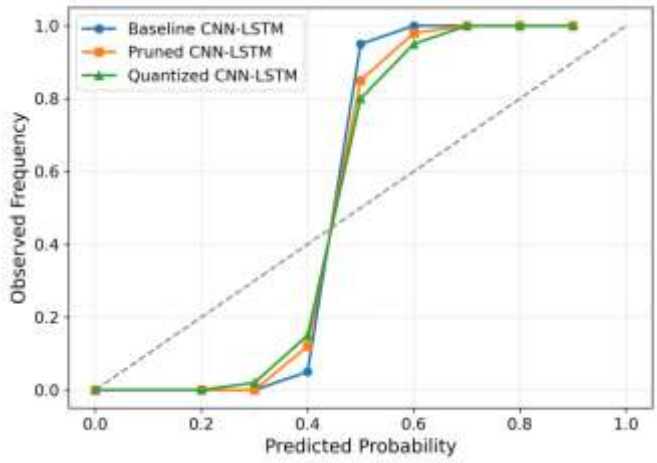


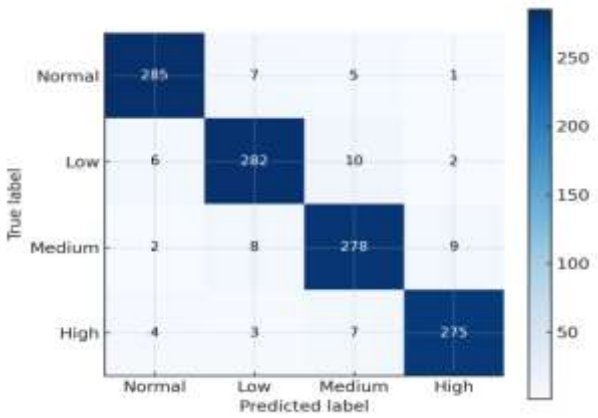
Figure 10. Calibration curves for model variants

The calibration curve in Figure 10 shows that all three models produce well-calibrated probability estimates, with their curves closely following the diagonal reference line. At low predicted probabilities (0.0-0.3), all models slightly underestimate risk, which is expected in datasets dominated by normal cases. In the mid-range (0.5-0.7), the observed frequencies rise sharply, indicating that predicted probabilities accurately correspond to actual risk levels. At higher probabilities (0.8-1.0), the baseline CNN-LSTM aligns most closely with the ideal line, while the pruned and quantized models show only minor deviation. These results confirm that compression minimally affects calibration quality and that all three models provide reliable probability estimates for cardiovascular risk prediction.

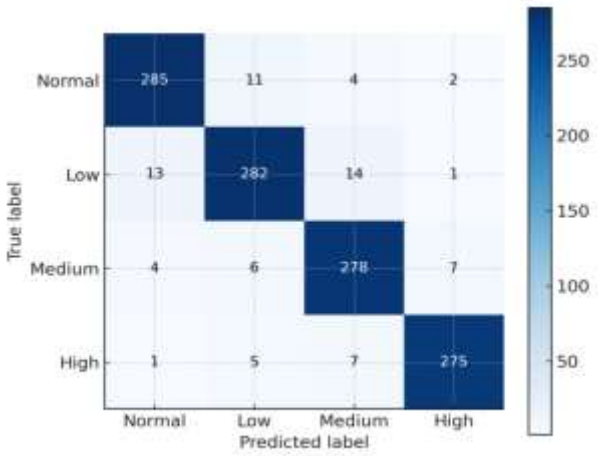
The confusion matrices in Figures 11(a)-(c) reflect classification performance across the four risk classes. Each model demonstrates high diagonal concentrations, with correctly identified samples. Off-diagonal entries remain comparatively low, typically within the range of 1 to 15 samples, indicating limited dispersion of predictions across neighboring categories.

Table 7 highlights how memory usage, inference time, and power consumption vary across the three model stages. The baseline CNN-LSTM model is the most resource-intensive, requiring 610MB of memory and consuming over 16 watts of power. Pruning substantially reduces these requirements, cutting memory needs to 370MB and decreasing power usage by almost 40%. Quantization further optimizes the model, reducing memory to 190MB and lowering power consumption to below 8 watts. These optimizations make the model feasible for deployment on energy-limited wearable devices.

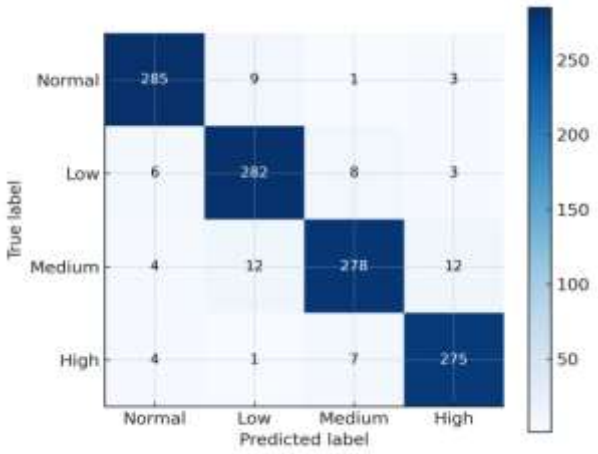
The CNN-LSTM model was trained using the Adam optimizer with a learning rate of 0.001, batch size of 32, and default momentum parameters (β_1=0.9, β_2=0.999) with ε=1e-8. Training was performed for a maximum of 100 epochs using the categorical cross-entropy loss function. To prevent overfitting and ensure stable convergence, an early-stopping strategy was applied with a patience of 10 epochs and a minimum required improvement of 0.001 in validation loss. Additionally, model checkpointing was enabled to automatically save the best-performing model based on validation accuracy during training, ensuring that the final deployed model corresponded to the optimal epoch.



(a)



(b)



(c)

Figure 11. (a) Confusion matrix- CNN-LSTM baseline model (b) Confusion matrix- CNN-LSTM pruned model (c) Confusion matrix- CNN-LSTM pruned and quantized model model

Table 7. Resource efficiency comparison

Model Version	Memory (MB)	Inference Time (s)	Power Consumption (W)
Baseline CNN-LSTM	610	71	16.74
Pruned CNN-LSTM	370	50	10.20
Quantized CNN-LSTM	190	48	7.90

Table 8. Effect of batch size on performance (Pruned model)

Batch Size	Accuracy (%)	F1-Score (%)	Inference Time (s)	Memory (MB)
16	92.2	92	53	365
32	95.27	93.3	50	370
64	91.1	92.6	47	375
128	90.4	90	43	380

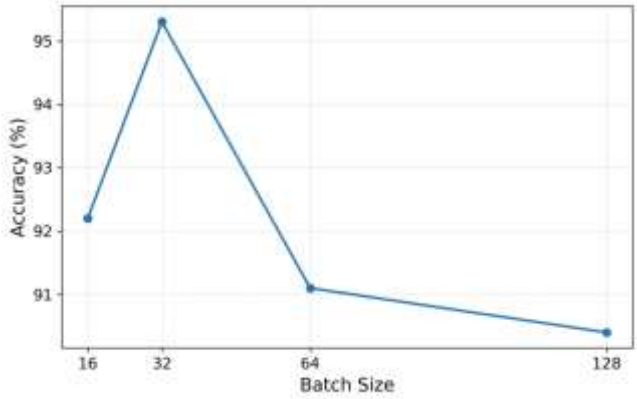


Figure 12. Accuracy vs. batch size

Table 8 and the corresponding plot in Figure 12 together analyze the impact of batch size on model performance and efficiency for the pruned model. A batch size of 32 achieves the highest accuracy (95.27%) and F1-score, indicating an optimal balance between stable gradient updates, memory utilization, and learning effectiveness. Increasing the batch size to 64 and 128 leads to a slight reduction in accuracy while improving inference speed, highlighting the typical trade-off between computational efficiency and learning stability in deep learning models. Overall, the results show that moderate batch sizes provide the best performance, whereas very large batch sizes, despite reducing training noise, result in marginally lower accuracy due to less frequent weight updates.

Table 9. Effect of learning rate on accuracy (Quantized model)

Learning Rate (LR)	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
0.1	88.4	88.1	88	88
0.01	91.3	90.9	91	90.8
0.001	92.3	91.9	92.1	92
0.0001	91	90.5	90.6	90.5

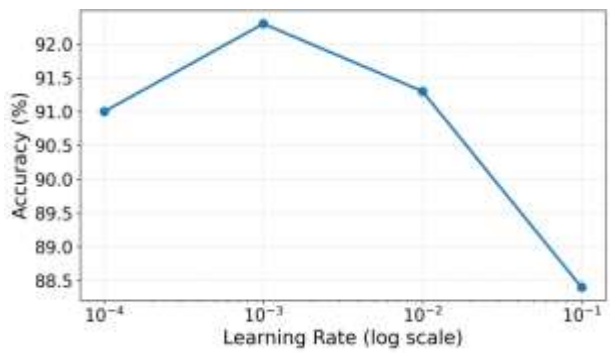


Figure 13. Accuracy vs learning rate (Quantized model)

Table 9 and Figure 13 together illustrate the effect of learning rate on model performance. A learning rate of 0.001

achieves the best accuracy, providing an optimal balance between stable convergence and effective learning. In contrast, a high learning rate (0.1) results in unstable training and reduced performance, while a very low learning rate (0.0001) slows convergence without meaningful accuracy improvements. These results highlight the importance of careful learning-rate tuning, even after model compression.

The results presented in Table 10 and Figure 14 demonstrate the impact of dropout on the baseline CNN-LSTM model's accuracy and generalization. A dropout rate of 0.2 provides the best balance, achieving 95.27% accuracy while effectively reducing overfitting. In contrast, the absence of dropout leads to slightly higher accuracy but increases overfitting risk, whereas excessive dropout (0.6) causes underfitting and degrades predictive performance. These results highlight the importance of moderate regularization for robust wearable healthcare models.

Table 10. Effect of dropout rate

Dropout Rate	Accuracy (%)	F1-Score (%)	Overfitting Observed
0.0	96	95.7	Yes
0.2	95.27	94.8	No
0.4	93.9	93.6	No
0.6	91.2	90.7	Slight underfitting

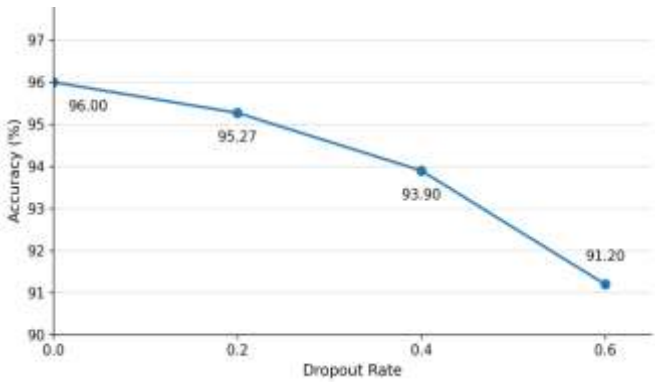


Figure 14. Accuracy vs dropout rate

Table 11 and Figure 15 illustrate the relationship between pruning percentages and the resulting accuracy and model size. As pruning levels increase from 30% to 90%, model size shrinks dramatically from 610MB to just 170MB—while accuracy gradually declines. Up to 50% pruning, the accuracy remains relatively stable (above 95%), demonstrating that substantial compression is possible without severely affecting model performance. Beyond 70% pruning, the model's predictive ability drops more noticeably.

Table 11. Pruning ratio vs accuracy and model size

Pruning (%)	Accuracy (%)	Model Size (MB)
0	95.27	610
30	95.12	490
50	95.15	370
70	95.15	290
90	88.2	170

Table 12 compares different quantization techniques: post-training int8 quantization, dynamic range quantization, and Float16 quantization. Post-training int8 quantization delivers the highest memory and energy savings, reducing model size to 190MB with only a minor decrease in accuracy. Float16 maintains slightly higher accuracy but requires more memory, indicating a trade-off between precision retention and deployment feasibility on low-power hardware.

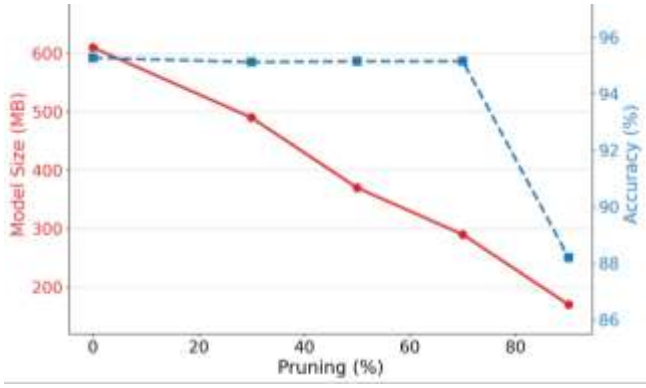


Figure 15. Pruning vs accuracy and model size

Table 12. Effect of quantizationtype

Quantization Type	Accuracy (%)	Inference Time (s)	Power (W)	Model Size (MB)
None (Float32)	95.27	71	16.74	610
Post-training int8	94.3	48	7.9	190
Dynamic Range	91.5	52	8.4	200
Float16	93.2	55	9.1	310

Table 13. Number of LSTM units vs accuracy

LSTM Units	Accuracy (%)	Memory Usage (MB)	Inference Time (s)
32	90.2	290	45
64	92.8	330	48
128	95.27	610	71
256	95.7	880	89

Insights from Table 13 and the corresponding trends in Figure 16 demonstrate the effect of increasing LSTM units on accuracy, memory consumption, and inference time. Adding more units improves accuracy, with peak performance (~95.7%) at 256 units. However, memory usage rises sharply with more units, from 290MB at 32 units to 880MB at 256 units. This trade-off is critical: 128 LSTM units offer a sweet spot, balancing high accuracy and reasonable memory requirements for wearable applications.

Table 14 presents a detailed evaluation of how combined pruning and quantization affect model performance, power consumption, inference time, and model size. Moderate pruning (30%-50%) coupled with quantization yields the best balance, sustaining accuracy above 94% while reducing power consumption and model size substantially. Heavier pruning (70%-90%) significantly cuts memory and energy costs but at the expense of noticeable drops in predictive performance, making it less ideal for critical healthcare applications where reliability is crucial.

The heatmap in Figure 17 illustrates how different levels of pruning combined with int8 quantization impact model performance metrics. As pruning levels increase from 30% to 90%, both model size and power consumption decrease sharply, making the models more suitable for edge devices. However, this compression comes at the cost of reduced accuracy, especially beyond 70% pruning. The heatmap highlights the importance of carefully choosing the pruning

threshold to maintain acceptable accuracy while maximizing resource efficiency.

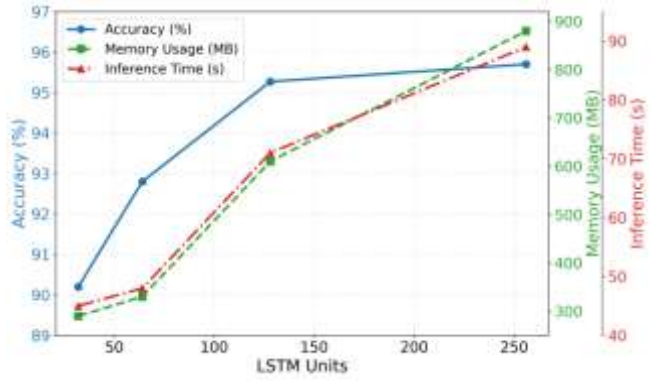


Figure 16. Accuracy and memory vs. number of LSTM Units

Table 14. Combined pruning + quantization levels impact

Pruning (%)	Quanti-zation	Accuracy (%)	Power (W)	Inference Time (s)	Model Size (MB)
30	int8	94.26	8.5	52	210
50	int8	94.28	7.9	48	190
70	int8	91.17	7.2	45	160
90	int8	84.8	6.5	42	130

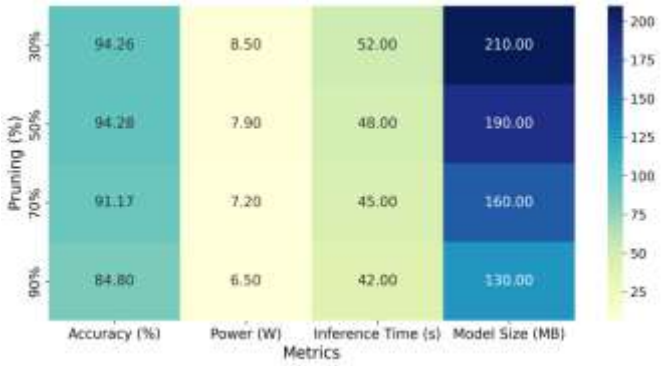


Figure 17. Optimization levels vs performance

The CNN-LSTM baseline model delivered the highest classification performance with an accuracy of 95. 27%, a precision of 92. 8% and an F1 score of 94.8%. However, it required 610MB of memory and consumed 16.74 watts of power, rendering it infeasible for deployment on low-power wearable devices. To overcome these limitations, TinyML optimization techniques were applied.

These results clearly demonstrate that the primary contribution of pruning is memory and power reduction, not accuracy improvement. The slight decrease in accuracy is a trade-off for significant computational gains, which is often acceptable in wearable applications where hardware constraints are critical. The interpretation of the work is that TinyML techniques can preserve diagnostic reliability while transforming otherwise heavy deep learning models into lightweight versions suitable for edge-level inference. In other words, the model does not become more accurate after optimization; it becomes more efficient while retaining sufficient accuracy for practical use in the real world.

Ultimately, the actual result and central insight of this research is the demonstration that a carefully optimized CNN-LSTM model can serve as a reliable, low-power, real-time diagnostic engine for wearable health monitoring devices.

5. CONCLUSION

In this study, we presented an energy-efficient deep learning approach for heart disease risk prediction using wearable sensor data, combining a hybrid CNN-LSTM architecture with TinyML optimization techniques. By leveraging the feature extraction capabilities of Convolutional Neural Networks and the sequential modeling strengths of Long Short-Term Memory networks, the proposed model effectively captured both spatial and temporal patterns within multimodal physiological signals.

Recognizing the constraints imposed by wearable devices such as limited memory, computation power, and battery life-we incorporated TinyML strategies, namely structured pruning and post-training quantization, to compress the model without significantly compromising predictive accuracy. Experimental evaluations demonstrated that the TinyML-optimized CNN-LSTM model achieved a competitive test accuracy of up to 94.28%, while reducing memory consumption by over 65% and lowering inference time and power requirements significantly compared to the baseline model.

These results validate that deep learning models, when appropriately optimized, can be deployed on resource-constrained edge devices to enable real-time, continuous, and personalized cardiovascular monitoring.

This research takes a significant step toward the realization of intelligent, low-power, and proactive healthcare monitoring systems accessible to a broader global population.

6. LIMITATION AND FUTURE SCOPE

Although the proposed model demonstrates strong predictive performance, several limitations must be acknowledged. First, wearable-derived physiological signals are inherently susceptible to motion artifacts, environmental noise, user compliance issues, and sensor-placement variability, all of which can affect measurement reliability. While preprocessing steps including artifact filtering, baseline correction, and signal-quality assessment were applied to reduce these effects, residual noise may still influence model predictions. Second, the risk categories used in this study, although aligned with known clinical thresholds, do not replace formal diagnostic evaluation. Therefore, the model is intended as ascreening and decision-support tool, not a standalone diagnostic system.Compression through pruning and quantization may introduce small shifts in sensitivity, so we verified calibration and error trends to ensure safety.

Future work will focus on prospective validation in clinical settings, conducted in collaboration with cardiovascular specialists to assess real-world utility and diagnostic impact. Additional improvements includeexpanding to multi-center datasets, incorporating device-specific calibration, developing personalized risk-adaptation strategies, and integrating confidence scoring for clinician oversight. Such efforts are essential before clinical deployment to ensure reliability, interpretability, and alignment with medical decision-making.

REFERENCES

[1] Roth, G.A., Mensah, G.A., Johnson, C.O., Addolorato, G., Ammirati, E., Baddour, L.M., GBD-NHLBI-JACC Global Burden of Cardiovascular Diseases Writing Group. (2020). Global burden of cardiovascular diseases and risk factors, 1990-2019: Update from the GBD 2019 study. Journal of the American college of cardiology, 76(25): 2982-3021. https://doi.org/10.1016/j.jacc.2020.11.010

[2] Johnson, A.E., Bulgarelli, L., Shen, L., Gayles, A., Shammout, A., Horng, S., Mark, R.G. (2023). MIMIC-IV, a freely accessible electronic health record dataset. Scientific data, 10(1): 1. https://doi.org/10.1038/s41597-022-01899-x

[3] Bhatt, C.M., Patel, P., Ghetia, T., Mazzeo, P.L. (2023). Effective heart disease prediction using machine learning techniques. Algorithms, 16(2): 88. https://doi.org/10.3390/a16020088

[4] Siirtola, P., Koskimäki, H., Mönttinen, H., Röning, J. (2018). Using sleep time data from wearable sensors for early detection of migraine attacks. Sensors,18(5): 1374. https://doi.org/10.3390/s18051374

[5] Martin-Gonzalez, S., Navarro-Mesa, J.L., Julia-Serda, G., Kraemer, J.F., Wessel, N., Ravelo-Garcia, A.G. (2017). Heart rate variability feature selection in the presence of sleep apnea: An expert system for the characterization and detection of the disorder. Computers in Biology and Medicine, 91: 47-58. https://doi.org/10.1016/j.compbiomed.2017.10.004

[6] Singhal, A., Cowie, M.R. (2020). The role of wearables in heart failure. Current Heart Failure Reports, 17(4): 125-132. https://doi.org/10.1007/s11897-020-00467-x

[7] Perez-Pozuelo, I., Posa, M., Spathis, D., Westgate, K., Wareham, N., Mascolo, C., Brage, S., Palotti, J. (2022). Detecting sleep outside the clinic using wearable heart rate devices. Scientific Reports, 12(1): 7956. https://doi.org/10.1038/s41598-022-11792-7

[8] Alday, E.A.P., Gu, A., Shah, A.J., Robichaux, C., Wong, A.K.I., Liu, C., Rad, A.B., Elola, A., Seyedi, S., Li, Q., Sharma, A., Clifford, G.D., Reyna, M.A. (2020). Classification of 12-lead ECGs: The physionet / computing in cardiology challenge 2020. Physiological Measurement, 41(12): 124003. https://doi.org/10.1088/1361-6579/abc960

[9] Lin, J., Fu, R., Zhong, X., Yu, P., Tan, G., Li, W.,Zhou, L., Ning, C. (2021). Wearable sensors and devices for real-time cardiovascular disease monitoring. Cell Reports Physical Science, 2(8): 100541. https://doi.org/10.1016/j.xcrp.2021.100541

[10] De Zambotti, M., Baker, F.C., Willoughby, A.R., Godino, J.G., Wing, D., Patrick, K., Colrain, I.M. (2016). Measures of sleep and cardiac functioning during sleep using a multi-sensory commercially-available wristband in adolescents. Physiology & Behavior, 158: 143-149. https://doi.org/10.1016/j.physbeh.2016.03.006

[11] Sarmah, S.S. (2020). An efficient IoT-based patient monitoring and heart disease prediction system using deep learning modified neural network. IEEE Access, 8: 135784-135797. https://doi.org/10.1109/ACCESS.2020.3007561

[12] Ali, F., El-Sappagh, S., Islam, S.R., Kwak, D., Ali, A., Imran, M., Kwak, K.S. (2020). A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. Information Fusion, 63: 208-222. https://doi.org/10.1016/j.inffus.2020.06.008

[13] Kundrick, J., Naniwadekar, A., Singla, V., Kancharla, K., Bhonsale, A., Voigt, A., Saba, S. (2025). Machine learning applied to wearable fitness tracker data and the risk of hospitalizations and cardiovascular events. American Journal of Preventive Cardiology, 22: 101006. https://doi.org/10.1016/j.ajpc.2025.101006

[14] Deperlioglu, O., Kose, U., Gupta, D., Khanna, A., Sangaiah, A.K. (2020). Diagnosis of heart diseases by a secure internet of health things system based on autoencoder deep neural network. Computer Communications, 162: 31-50. https://doi.org/10.1016/j.comcom.2020.08.011

[15] Pakhomov, S., Weston, S.A., Jacobsen, S.J., Chute, C.G., Meverden, R., Roger, V.L. (2007). Electronic medical records for clinical research: Application to the identification of heart failure. American Journal of Managed Care, 13(6): 281-288.

[16] García-Ordás, M.T., Bayón-Gutiérrez, M., Benavides, C., Aveleira-Mata, J., Benítez-Andrades, J.A. (2023). Heart disease risk prediction using deep learning techniques with feature augmentation. Multimedia Tools and Applications, 82(20): 31759-31773. https://doi.org/10.1007/s11042-023-14817-z

[17] Alqurashi, F., Zafar, A., Khan, A.I., Almalawi, A., Alam, M.M., Azim, R. (2023). Deep neural network and predator crow optimization-based intelligent healthcare system for predicting cardiac diseases. Mathematics, 11(22): 4621. https://doi.org/10.3390/math11224621

[18] Mohammad, F., Al-Ahmadi, S. (2023). WT-CNN: A hybrid machine learning model for heart disease prediction. Mathematics, 11(22): 4681. https://doi.org/10.3390/math11224681

[19] Shankar, V., Kumar, V., Devagade, U., Karanth, V., Rohitaksha, K. (2020). Heart disease prediction using CNN algorithm. SN Computer Science, 1(3): 170. https://doi.org/10.1007/s42979-020-0097-6

[20] Ram Kumar, R.P., Raju, S., Annapoorna, E., Hajari, M., Hareesa, K., Vatin, N.I., Joshi, A., AL-Attabi, K. (2024). Enhanced heart disease prediction through hybrid CNN-TLBO-GA optimization: A comparative study with conventional CNN and optimized CNN using FPO algorithm. Cogent Engineering, 11(1): 2384657. https://doi.org/10.1080/23311916.2024.2384657

[21] Subashini, A., Kanaka Raju, P. (2023). An IoT-based heart disease diagnosis system using gradient boosting and deep convolution neural network. SN Computer Science, 5(1), 23. https://doi.org/10.1007/s42979-023-02340-9

[22] Abutalip, S., Baikuvekov, M., Zanggar, D. (2024). A machine learning approach for detection of heart diseases using wearable devices. In 2024 IEEE 4th International Conference on Smart Information Systems and Technologies (SIST), Astana, Kazakhstan, pp. 266-271. https://doi.org/10.1109/SIST61555.2024.10629367

[23] Al Reshan, M.S., Amin, S., Zeb, M.A., Sulaiman, A., Alshahrani, H., Shaikh, A. (2023). A robust heart disease prediction system using hybrid deep neural networks. IEEE Access, 11: 121574-121591. https://doi.org/10.1109/ACCESS.2023.3328909

[24] Xia, B., Innab, N., Kandasamy, V., Ahmadian, A., Ferrara, M. (2024). Intelligent cardiovascular disease

diagnosis using deep learning enhanced neural network with ant colony optimization. Scientific Reports, 14(1): 21777. https://doi.org/10.1038/s41598-024-71932-z

[25] Arooj, S., Rehman, S.U., Imran, A., Almuhaimeed, A., Alzahrani, A.K., Alzahrani, A. (2022). A deep convolutional neural network for the early detection of heart disease. Biomedicines, 10(11): 2796. https://doi.org/10.3390/biomedicines10112796

[26] Neri, L., Oberdier, M.T., Van Abeelen, K.C., Menghini, L., Tumarkin, E., Tripathi, H., Jaipalli, S., Orro, A., Paolocci, N., Gallelli, Il., Dall'Olio, M., Beker, A., Carrick, R.T., Borghi, C., Halperin, H.R. (2023). Electrocardiogram monitoring wearable devices and artificial-intelligence-enabled diagnostic capabilities: A review. Sensors, 23(10): 4805. https://doi.org/10.3390/s23104805

[27] Qureshi, F., Krishnan, S. (2018). Wearable hardware design for the internet of medical things (IoMT). Sensors, 18(11): 3812. https://doi.org/10.3390/s18113812

[28] Sun, B., Bayes, S., Abotaleb, A.M., Hassan, M. (2023). The case for tinyML in healthcare: CNNs for real-time on-edge blood pressure estimation. In Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, pp. 629-638. https://doi.org/10.1145/3555776.3577747

[29] Mahardika T.N.Q., Fuadah, Y.N., Jeong, D.U., Lim, K.M. (2023). PPG signals-based blood-pressure estimation using grid search in hyperparameter optimization of CNN-LSTM. Diagnostics, 13(15): 2566. https://doi.org/10.3390/diagnostics13152566

[30] Arthi, R., Krishnaveni, S. (2024). Optimized tiny machine learning and explainable AI for trustable and energy-efficient fog-enabled healthcare decision support system. International Journal of Computational Intelligence Systems, 17(1): 229. https://doi.org/10.1007/s44196-024-00631-4.

[31] Elhanashi, A., Dini, P., Saponara, S., Zheng, Q. (2024). Advancements in TinyML: Applications, limitations, and impact on IoT devices. Electronics, 13(17): 3562. https://doi.org/10.3390/electronics13173562

[32] Greenacre, M., Groenen, P.J., Hastie, T., d'Enza, A.I., Markos, A., Tuzhilina, E. (2022). Principal component analysis. Nature Reviews Methods Primers, 2(1): 100. https://doi.org/10.1038/s43586-022-00184-w

[33] Fernández, A., Garcia, S., Herrera, F., Chawla, N.V. (2018). SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. Journal of Artificial Intelligence Research, 61: 863-905. https://doi.org/10.1613/jair.1.11192

[34] Colantonio, L.D., Booth, J.N., Bress, A.P., Whelton, P.K., Shimbo, D., Levitan, E.B., Howard, G., Safford, M.M., Muntner, P. (2018). 2017 ACC/AHA blood pressure treatment guideline recommendations and cardiovascular risk. Journal of the American College of Cardiology, 72(11): 1187-1197. https://doi.org/10.1016/j.jacc.2018.05.074

[35] Jarczok, M.N., Weimer, K., Braun, C., Williams, D.P., Thayer, J.F., Guendel, H.O., Balint, E.M. (2022). Heart rate variability in the prediction of mortality: A systematic review and meta-analysis of healthy and patient populations. Neuroscience & Biobehavioral Reviews, 143: 104907. https://doi.org/10.1016/j.neubiorev.2022.104907