18844- Final demo

Off the cuff : Blood Pressure Monitoring Device

Om Kulkarni
Pradhumna Guru Prasad
Aidan Vogt

Carnegie Mellon University

Abstract

The problem

Current state:
BP monitoring is episodic
and intrusive (cuff)

The gap:
Hypertension is dynamic. A
snapshot once in a while
misses the full picture.

Our solution:
An embedded DL enabled device with the capability of continuous, passive monitoring using a single optical sensor.

Why Embedded ML?

B

Bandwidth:
We don't stream raw data
we only output the final BP value.

L

Latency:
Instant Feedback.
Processes 100Hz local data for real-time display

E

Energy:
Optimized for low-power micro-controllers..

R

Reliability:
Works fully offline (vital for medical devices).

P

Privacy:
Raw biometric data never leaves the device.

Evolution (Approach)

Phase 1: Beat-Level LSTM

Idea:
Analyze individual heartbeats using a Long Short-Term Memory network.

Verdict:
Failed
Too susceptible to noise and lacked sufficient context for accurate predictions. (Mean Absolute Error ~21 mmHg).

Phase 2: Pulse Transit Time (PTT)

Idea:
Correlate the timing difference between ECG and PPG signals to derive blood pressure.

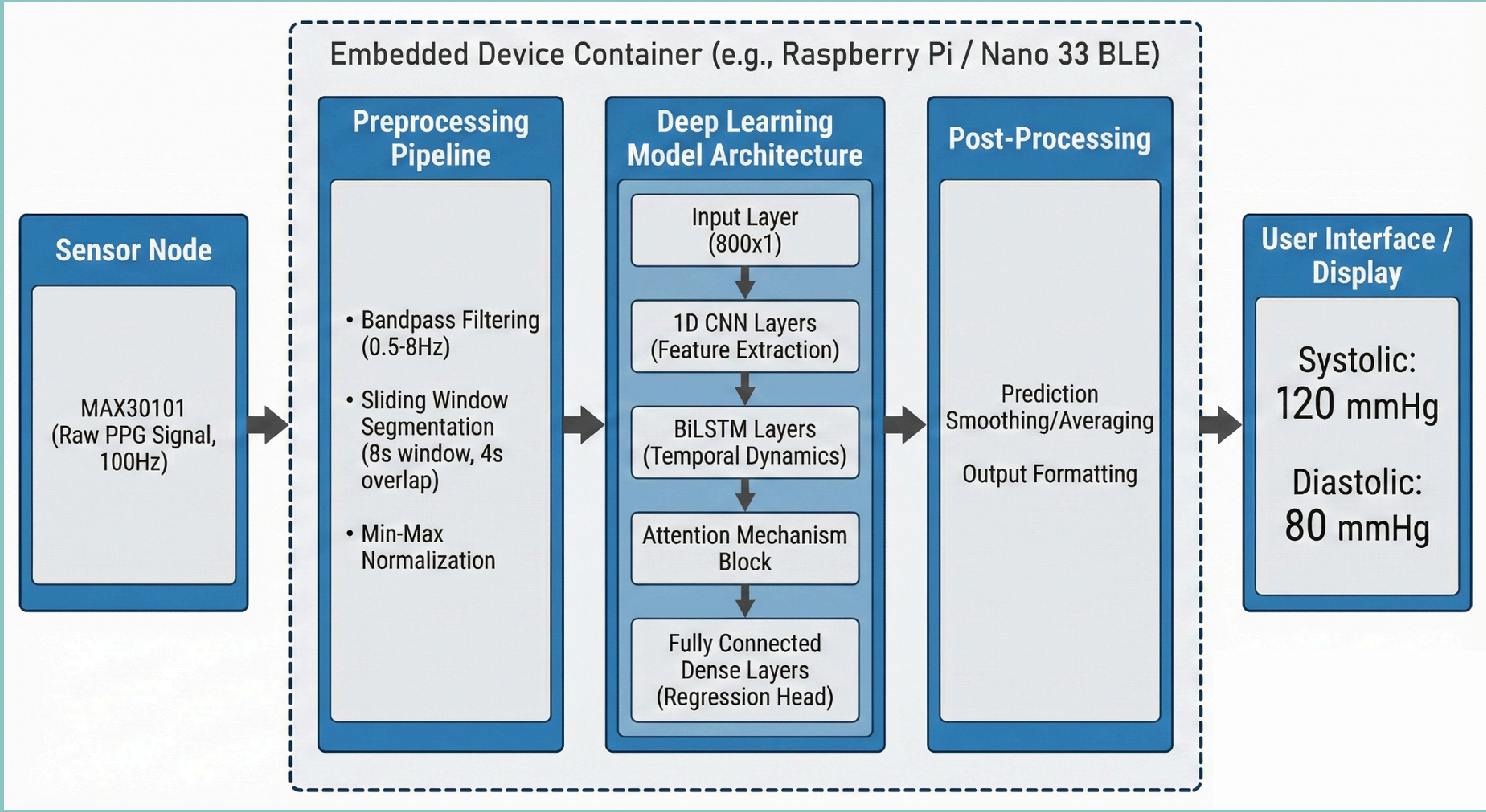
Verdict:
Accurate but Impractical.
Required two distinct sensors (e.g., chest ECG and finger PPG), making it unsuitable for a single-sensor, wearable device.

Phase 3: Window-Level Deep Learning

Idea:
Analyze 8-second windows of photoplethysmography (PPG) blood flow signals.

Verdict:
Success
This approach allowed the model to identify and leverage trends within the data, effectively filtering out transient noise. This became the foundation for our current robust solution.

System Block Diagram



The AI engine (Model and preprocessing)

The Signal Pipeline (Preparing the Data)

Resampling: upscaled sensor data (100Hz) to 125Hz to match MIMIC training data.

↓

Filtering: Butterworth Band-pass (0.5–8 Hz) to remove motion artifacts and high-freq. noise.

↓

Windowing: 8-second sliding windows with a 4-second hop (Updates every 4s).

↓

Polarity Inversion: (flipped the signal) to match the peak direction of clinical PPG.

The Architecture (CNN-BiLSTM-Attention)

Feature Extractor

1D CNN extracts morphological shape features (slopes, dicrotic notch).

→

Temporal Context

BiLSTM analyzes the history of the wave (past & future context) to determine stability.

→

Additive Attention Mechanism

weights specific parts of the signal, allowing the model to focus on valid beats and ignore noise.

→

Output:

Joint Regression Head predicts SBP and DBP simultaneously.

Methods – Embedded Implementation

Hardware: Raspberry Pi Zero.

Sensor: MAX30101 (Reflective PPG).

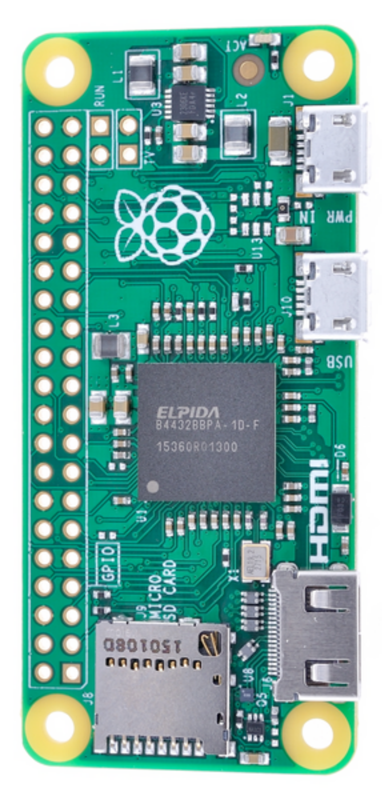
- Configured in IR-only mode (High sensitivity).
- 18-bit ADC resolution for detecting micro-changes in blood volume.

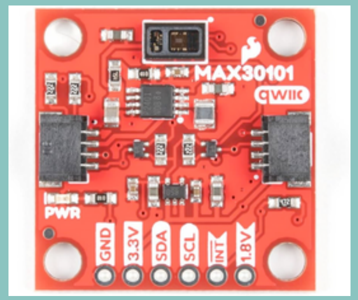
The Inference Loop (Python Pipeline)

- Acquisition: Streams Raw IR, Red, and Green data.
- Auto-Preprocessing: Real-time signal cleaning and Z-score normalization.
- Inference: The CNN-BiLSTM model runs on the live 8-second window.
- Result: Displays Per-window BP estimates.

Performance Status

- **Latency**: System runs in Near Real-Time.
- **Morphology**: Live sensor waveforms successfully match the shape of the MIMIC training data (validating our preprocessing).

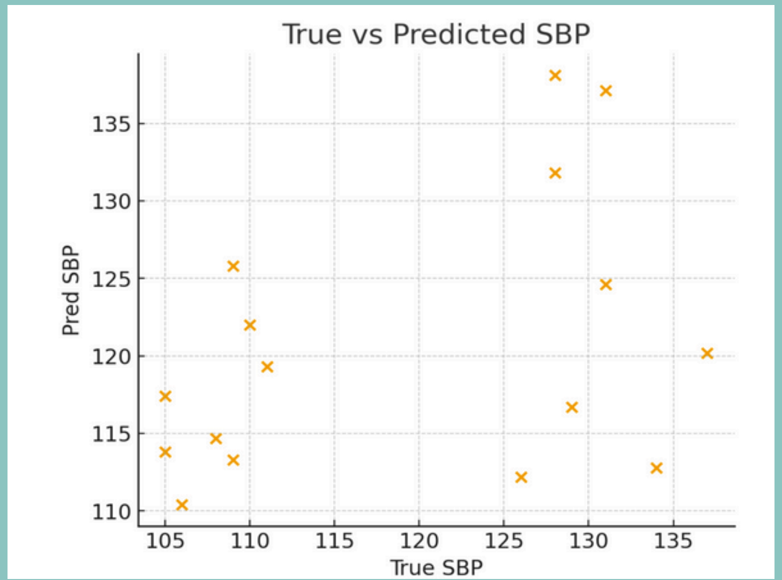


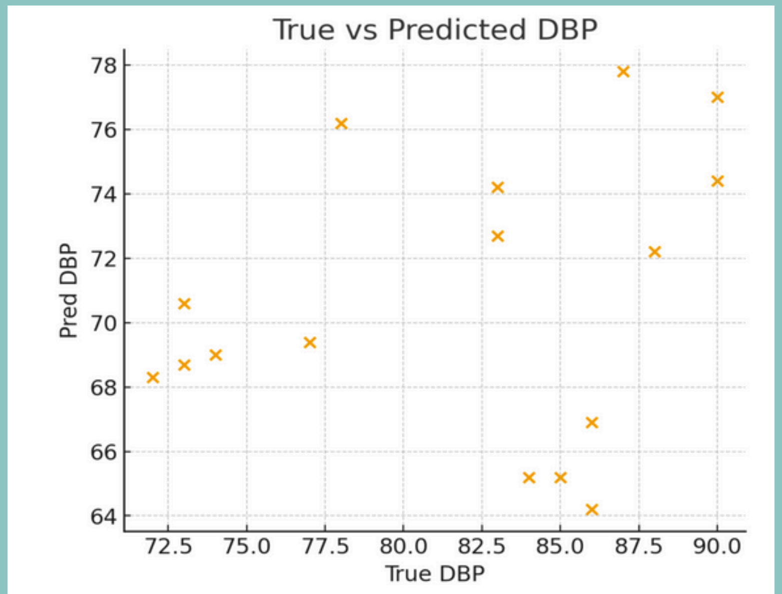


Carnegie Mellon University

Results – Accuracy & Validation

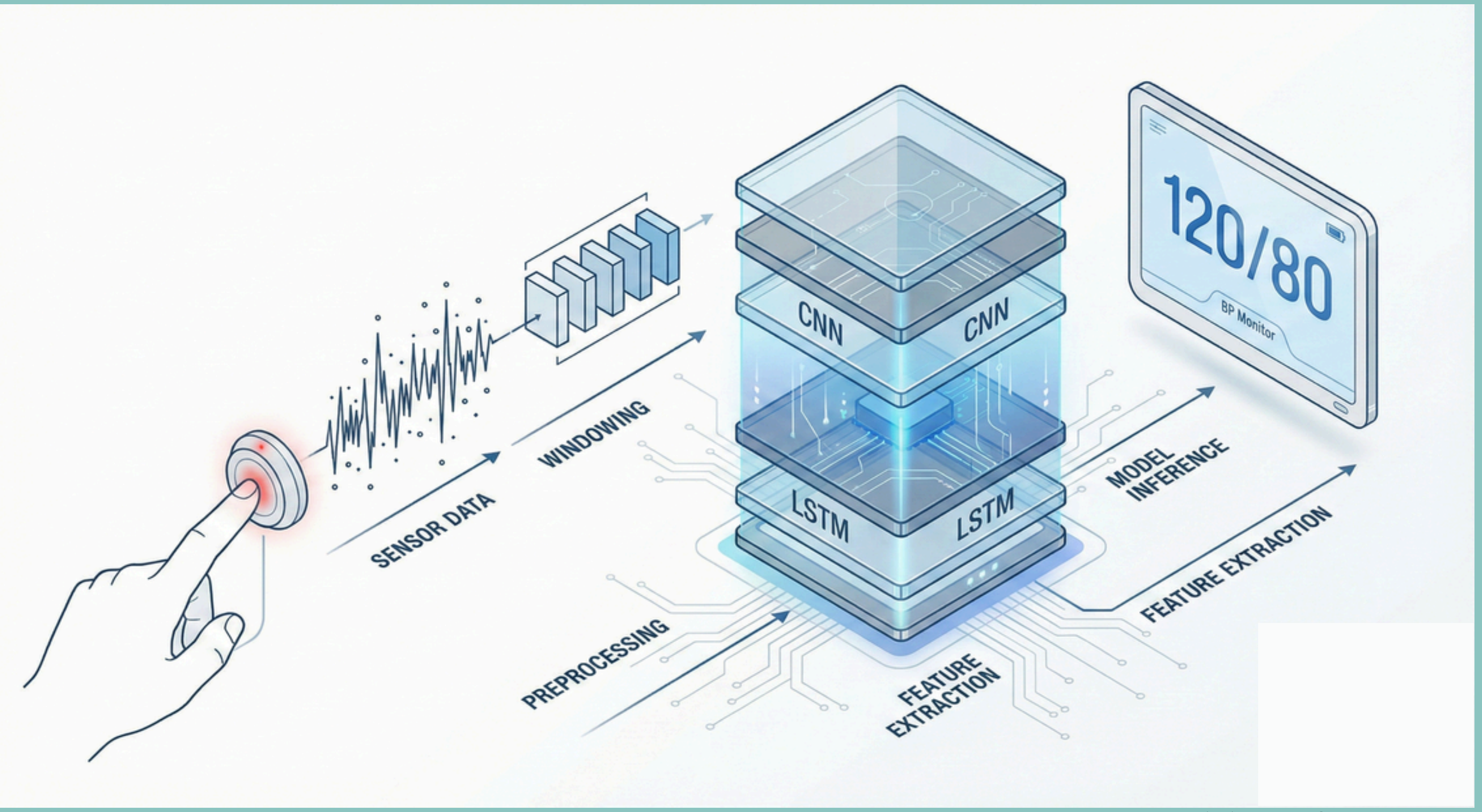
Results (Test dataset)	Hist Gradient Boosting regressor	beat level LSTM	CNN (PPG+ECG dataset)	Window level CNN (PPG only)	CNN+Bi-LSTM+Attention
MAE(Systolic BP)	14.69	21	17.83	9	7.8
RMSE(Systolic BP)	19	25.6	23.74	12.2	11
MAE(Diastolic BP)	7.45	8.8	8.19	4.5	4
RMSE(Diastolic BP)	10.11	11.7	11.89	6.8	6.3





Results (Live)	SBP	DBP
MAE (30 readings)	10.6	12.8
Mean bias	-3.1	-12

Live Demo



Source: Gemini nano banana pro

#	Start t (s)	SBP (mmHg)	DBP (mmHg)
1	0.0	118.4	73.1
2	4.0	120.9	74.4
3	8.0	132.4	77.6
4	12.0	138.9	75.5
5	16.0	130.3	71.5
6	20.0	113.5	65.5

Challenges and future work

Data Challenges:
Public datasets (MIMIC) are from sedated hospital patients; real-world users move around (Motion Artifacts).

The "Calibration Problem":
Deep learning learns general population trends but struggles with individual offsets (hence the DBP bias).

Motion Artifacts:
Hand movement destroys the optical signal. (Requires stationary measurement).

Hardware Challenges:
Sensitive to contact pressure (pressing too hard cuts off blood flow, too light creates noise).

Future Work:

- **Sensor Fusion:** Add Accelerometer (IMU) to cancel out motion noise.
- **Calibration:** Implement a "one-time calibration" step to fix the DBP offset.

References

Datasets

- MIMIC-III: Johnson, A. E. W., et al. (2016). "MIMIC-III, a freely accessible critical care database." Scientific Data, 3, 160035. (Accessed via PhysioNet).
- UCI Dataset: Kachuee, M., Kiani, K., Mohammadzade, H., & Shabany, M. (2015). "Cuff-less Blood Pressure Estimation Dataset." UCI Machine Learning Repository.

Key Methodology Papers

- Eom, H., et al. (2020). "End-to-End Deep Learning Architecture for Continuous Blood Pressure Estimation Using Attention Mechanism." Sensors, 20(8), 2338.
- Chen, W., et al. (2019). "Cuffless Blood Pressure Estimation Using Deep Learning with Photoplethysmography Signals." IEEE Access.
- Mukkamala, R., et al. (2015). "Toward Ubiquitous Blood Pressure Monitoring via Pulse Transit Time: Theory and Practice." IEEE Transactions on Biomedical Engineering.

QNA