AI-BASED REAL-TIME PATIENT TRIAGE AND MONITORING SYSTEM

Real-Time Clinical Severity Assessment via Multimodal Telemetry Fusion at the Edge

A multimodal AI prototype fusing ECG and chest X-ray analysis for edge-deployable, real-time patient triage.

Prof. Suman Chakraborty
Prof. Priyam Chakraborty

Dhritish Mondal
Arghyajyoti Ghosh

THE PROBLEM

Triage Decisions Are Time-Critical and Data-Fragmented

- Frontline clinical staff must rapidly assess cardiac and pulmonary risk from multiple independent data sources — ECG traces and chest X-rays — often under time pressure.
- ECG interpretation and chest X-ray review are typically performed separately, with no automated step that combines both signals into a single, actionable severity level.
- Manual, sequential review of each modality delays the identification of high-risk patients who need immediate escalation.
- Resource-constrained settings need a lightweight, edge-deployable system that can assist — not replace — the clinical decision-making process.

ECG review
Rhythm interpreted independently

↓

X-ray review
Imaging interpreted independently

No automated fusion step → delayed, inconsistent triage

MOTIVATION

Why an Edge-Deployable, Multimodal Approach

Real-time need

Cardiac and respiratory deterioration can progress quickly; triage support should operate continuously and in real time, not in batch.

Multimodal signal

ECG and chest X-ray capture different physiological information; combining them can surface risk patterns that either signal alone may miss.

Edge AI hardware

Low-cost accelerators such as the Raspberry Pi AI HAT+ (Hailo) make on-device inference feasible outside centralized hospital IT infrastructure.

Event-driven design

A messaging backbone (NATS) allows sensors, AI models, and dashboards to be decoupled and scaled independently.

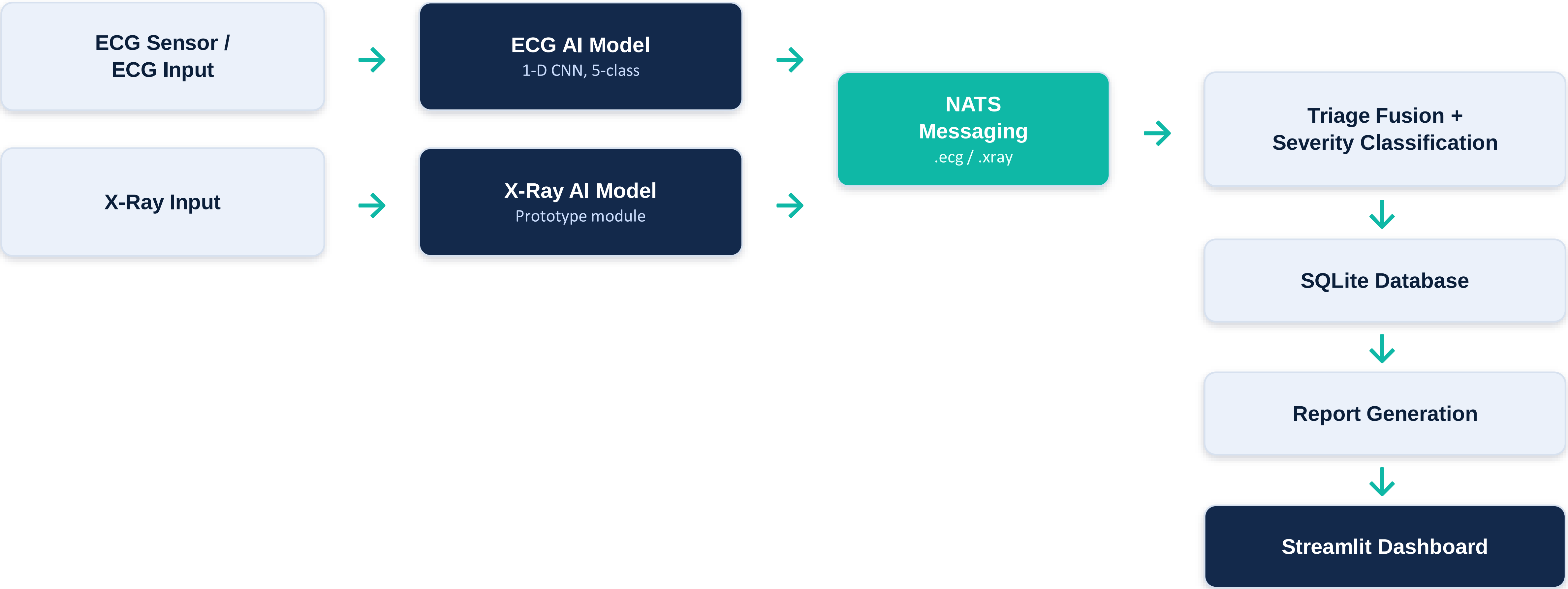
PROJECT OBJECTIVES

Eleven Integrated System Components

1. ECG analysis (deep learning classifier)
2. Chest X-ray analysis module
3. AI-based severity assessment / fusion
4. Real-time messaging using NATS
5. Database storage using SQLite
6. Automatic patient report generation
7. Streamlit monitoring dashboard
8. System performance monitoring
9. Raspberry Pi 5 edge deployment
10. Raspberry Pi AI HAT+ acceleration
11. Future real ECG sensor integration

SYSTEM DESIGN

Overall Architecture



Two independent AI modalities are fused at the patient level into a single, actionable severity output.

SYSTEM DESIGN

End-to-End Data Flow

Publisher (ECG / X-ray msg) → NATS Broker → Subscriber → AI Processing (ECG + X-ray) → Patient-Level Fusion → SQLite → Report → Dashboard

The pipeline is fully event-driven: each stage reacts to a message rather than polling for new data.

- Publishers simulate / forward ECG and X-ray patient data onto NATS subjects .ecg and .xray.
- Subscribers consume messages asynchronously and route them to the relevant AI model.
- Once both modalities for a given patient ID are available, patient-level fusion produces a single severity label.
- The result is persisted to SQLite, a text report is generated, and the dashboard reflects the update on its next refresh.

ECG Dataset

- Source: Kaggle “ECG Heartbeat Categorization Dataset” (author: Shayan Fazeli), MIT-BIH-derived.
- Files used for the current CNN training: mitbih_train.csv and mitbih_test.csv.
- Each row = one heartbeat: 187 ECG signal samples + 1 class label = 188 values per row.
- Five heartbeat classes are labeled, following the standard MIT-BIH annotation scheme.

Class	Label
0	Normal beat
1	Supraventricular ectopic beat
2	Ventricular ectopic beat
3	Fusion beat
4	Unknown beat

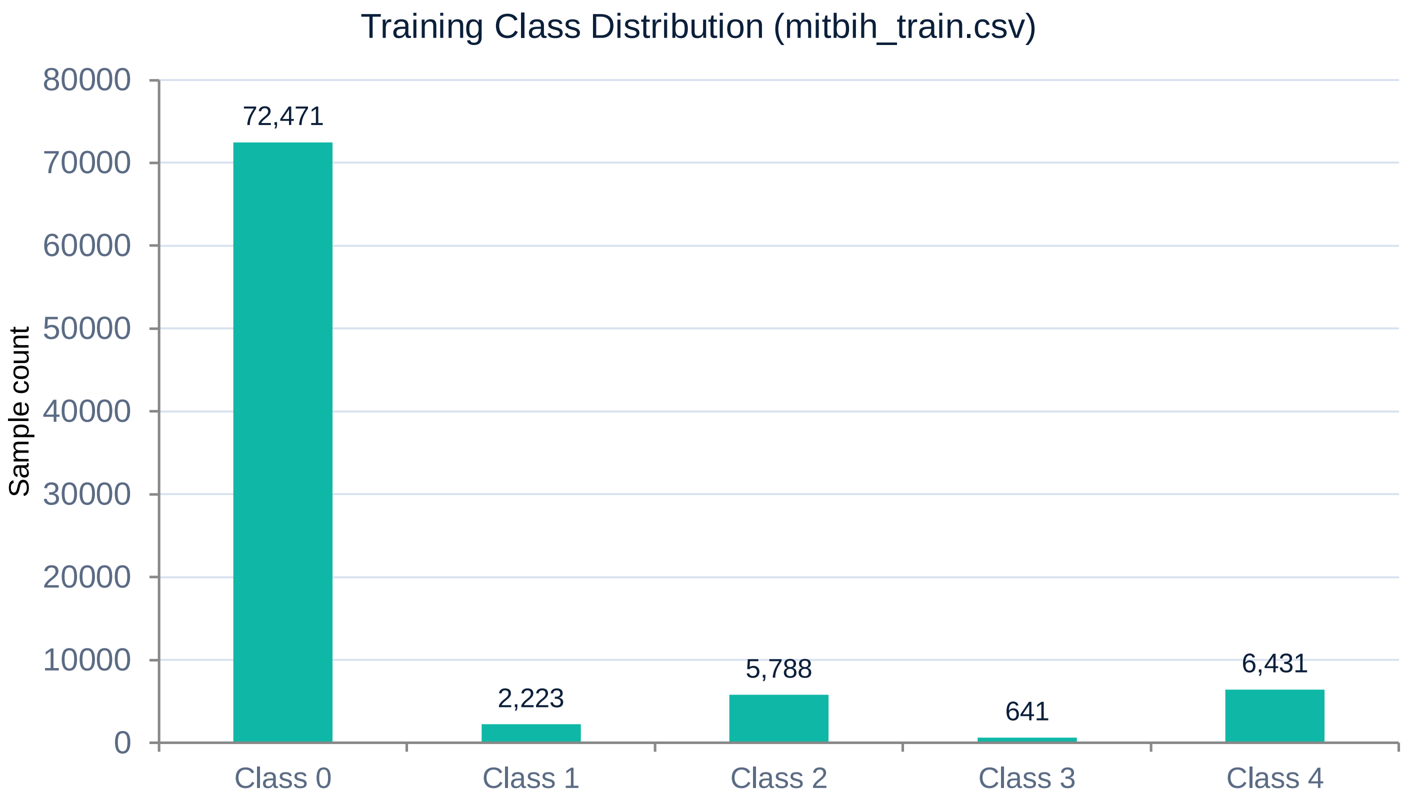
87,554
Training samples (approx.)

21,892
Testing samples (approx.)

OOSC

ECG Data Structure and Class Imbalance

Each training example: 187 samples (one heartbeat waveform) + 1 label → 188 columns per row.



~83%

of training data is Class 0 (Normal beat)

- The dataset is severely imbalanced: Class 0 alone accounts for roughly 83% of training examples.
- Rare classes (1 and 3 especially) have far fewer examples for the model to learn from.
- Accuracy alone is misleading here — a model that always predicts Class 0 would still score high accuracy while missing every abnormal beat.

ECG CNN Architecture

Input 187 samples → **Conv1D** 1→16 ch, k=5 → **ReLU** → **MaxPool1D** pool=2 → **Flatten** 16×91 → **FC** 1456→64 → **ReLU** → **FC** 64→5 classes

- ECG is a one-dimensional time-series signal — Conv1D is a natural fit for local waveform pattern detection.
- Convolution + pooling extract and downsample local morphological features (P-wave, QRS complex, T-wave shape).
- Two fully-connected layers map the extracted feature vector to a 5-class heartbeat prediction.

Training Configuration

Framework	PyTorch
Optimizer	Adam
Learning rate	0.001
Batch size	128
Epochs	10
Loss function	CrossEntropyLoss

Training Methodology

- The 5-class CNN was trained from scratch on mitbih_train.csv using PyTorch, with evaluation on the held-out mitbih_test.csv split.
- Adam optimizer with a fixed learning rate of 0.001 balances convergence speed with training stability.
- Mini-batches of 128 heartbeats were used per gradient step across 10 training epochs.
- CrossEntropyLoss was used for multi-class classification across the 5 heartbeat categories.
- Training ran on CPU or GPU depending on availability; no hardware-specific timing claims are made.

Optimizer	**Adam**
Learning Rate	**0.001**
Batch Size	**128**
Epochs	**10**
Loss	**CrossEntropyLoss**

Training Loss

233.8 → 64.9

Loss, epoch 1 → 10

- Loss decreases smoothly and monotonically across all 10 epochs with no signs of divergence.
- The steepest drop occurs in the first 2–3 epochs; later epochs show diminishing returns.
- Only the reported training loss and held-out test results are available — no separate validation curve was tracked during this run.

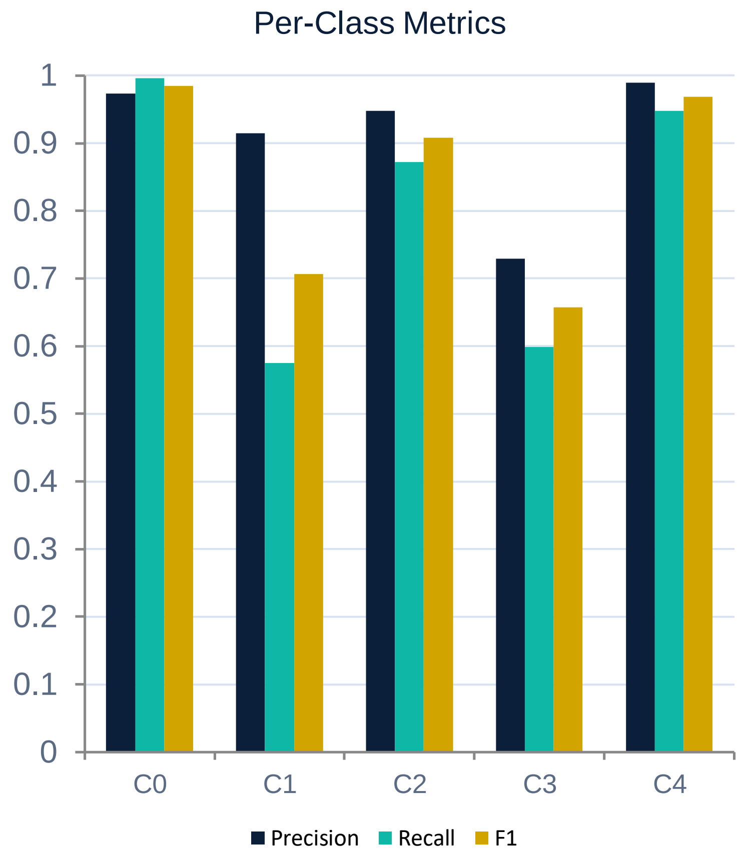
ECG Test Results

97.06%
Overall accuracy

0.8450
Macro-avg F1

0.9689
Weighted-avg F1

Class	Precision	Recall	F1	Support
Class 0 – Normal	0.9735	0.9960	0.9846	18,118
Class 1 – Supraventricular	0.9143	0.5755	0.7064	556
Class 2 – Ventricular ectopic	0.9475	0.8722	0.9083	1,448
Class 3 – Fusion beat	0.7293	0.5988	0.6576	162
Class 4 – Unknown	0.9896	0.9478	0.9682	1,608



Confusion Matrix (Test Set)

Actual \ Pred	C0	C1	C2	C3	C4
C0	18045	26	38	3	6
C1	225	320	11	0	0
C2	139	3	1263	33	10
C3	54	0	11	97	0
C4	73	1	10	0	1524

Rows = actual class, columns = predicted class. Darker cells indicate a larger share of that row's predictions.

- Class 0 (18,045 / 18,118 correct) and Class 4 (1,524 / 1,608) are classified with high reliability.
- Class 1 shows 225 of 545 misclassified samples routed to Class 0 — rare beats are often confused with the dominant Normal class.
- Class 3 has the smallest support (162) and the most scattered error pattern of any class.

Analysis of ECG Results

- **97.06% accuracy alone is not sufficient evidence of a strong model — it must be read alongside macro F1 and per-class recall.**
- Class 0 (Normal) and Class 4 (Unknown) perform very strongly — F1 of 0.9846 and 0.9682 respectively.
- Class 2 (Ventricular ectopic) also performs well, with F1 = 0.9083.
- Class 1 (Supraventricular) recall is only 57.55% — the model misses over 40% of these clinically important beats.
- Class 3 (Fusion beat) recall is only 59.88%, on the smallest support in the dataset (162 samples).
- **Macro-average F1 (0.8450) weighs all five classes equally and is the more informative summary statistic here, precisely because the dataset is dominated by Class 0.**
- Rare-class recall remains the model's principal weakness and the main target for future improvement.

NATS Real-Time Architecture

Publisher → NATS (.ecg / .xray) → Subscriber → ECG / X-Ray AI Processing → Patient-Level Fusion → SQLite → Report → Dashboard

- **Why NATS:**
- Asynchronous communication decouples data producers from AI processing consumers.
- Enables a real-time, event-driven architecture rather than periodic polling.
- Simplifies future integration with physical ECG sensors as additional publishers.
- Patient buffers associate ECG and X-ray results by patient ID; once both modalities arrive, the patient is processed by the fusion stage.

Subject: .ecg

Subject: .xray

Fault Tolerance

If NATS is unavailable, the system automatically falls back to an offline simulation mode for continued testing of AI, DB, and report logic.

Multimodal ECG + X-ray Fusion

ECG result
NORMAL / ANOMALOUS →

X-ray result
NORMAL / ATTENTION / CRITICAL →

Patient ID Buffer
Results are associated by patient ID until both modalities are available →

Severity Fusion Engine
Rule-based combination → single triage level

- Each modality produces an independent status: the ECG model outputs NORMAL/ANOMALOUS; the X-ray module outputs NORMAL/ATTENTION/CRITICAL.
- Patient-level buffering ensures fusion only occurs once both ECG and X-ray results exist for the same patient ID.
- The fusion stage converts two independent model outputs into one actionable, patient-level triage decision (see next slide for the decision logic).

Severity Decision Logic

ECG anomalous AND X-ray critical? → Yes → **CRITICAL — Immediate referral required**

No ↓

ECG anomalous OR X-ray critical? → Yes → **HIGH — Urgent specialist review needed**

No ↓

X-ray attention flagged? → Yes → **MODERATE — Monitor, follow up within 48h**

No ↓

NORMAL — No immediate action required

- The most severe combination of signals always wins — rules are evaluated in order of decreasing severity.
- This converts two independent model outputs into a single, actionable triage category and recommended action.
- **This is a prototype rule-based fusion layer, not a clinically validated triage protocol.**

Database + Report Generation

SQLite — records.db • table: triage_reports

- patient_id
- timestamp
- mean_hr
- mean_rr
- sdnn
- rmssd
- pnn50
- ecg_status
- ecg_confidence
- xray_status
- xray_flagged
- overall
- action

- Persistent patient results across sessions, enabling historical review.
- Serves as the single data source powering the Streamlit dashboard.
- Provides traceability between raw AI outputs and generated reports.
- Forms the foundation for future analytics on triage patterns.

Auto-Generated Report

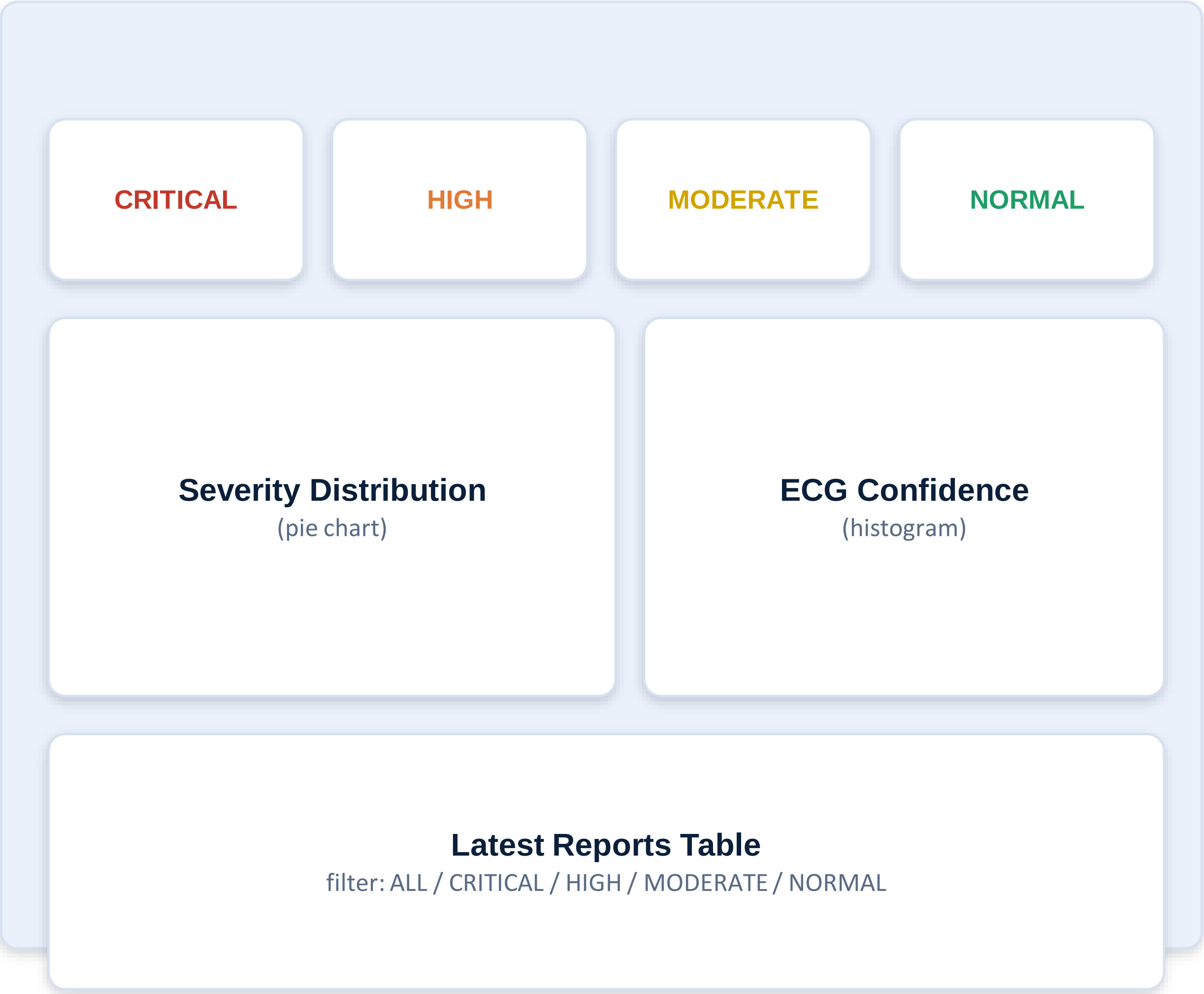
reports/PAT001_report.txt

```
 AI TRIAGE REPORT

Patient ID
Timestamp

ECG ANALYSIS
  Status
  Confidence

X-RAY ANALYSIS
  Status
  Findings

OVERALL TRIAGE
  Severity
  Action
```

MONITORING

Streamlit Dashboard

- System summary: total reports, and counts of Critical / High / Moderate / Normal cases.
- Severity distribution pie chart across all processed patients.
- ECG confidence histogram, with average, maximum, minimum and median confidence.
- Critical / High priority case table: patient ID, ECG status + confidence, X-ray status, overall severity, action.
- Severity filter (ALL / CRITICAL / HIGH / MODERATE / NORMAL) and a live view of the latest 10 reports.
- **Auto-refreshes every 5 seconds — functions as an operational monitoring interface, not just a static visualization.**



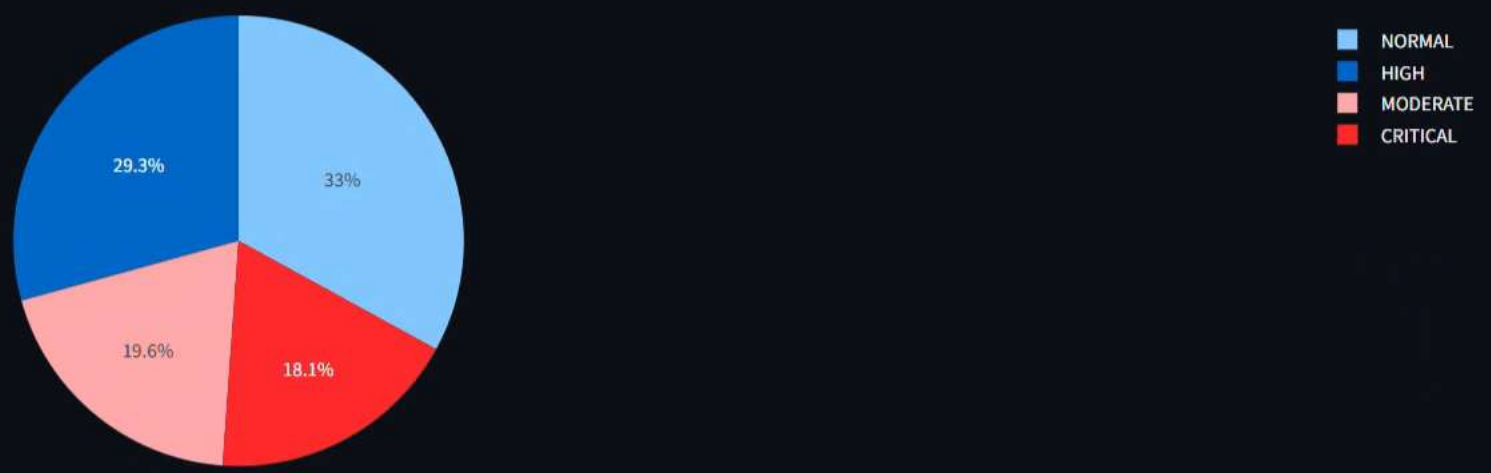
Deploy

System Summary

Total Reports	Critical	High	Normal	Moderate
485	88	142	160	95

Severity Distribution

Patient Severity Distribution



System Performance / Runtime Testing

The system also measures CPU usage, RAM usage, and per-patient process time during pipeline execution.

13% – 98%
CPU Usage

observed range across runs

78% – 85%
RAM Usage

observed range across runs

0.001s – 0.094s
Process Time

per-patient, observed range

- **These are prototype runtime observations, not fixed benchmark values.**
- Measured values depend heavily on system load, concurrent processes, and hardware at the time of the run.
- No formal, controlled benchmarking (e.g. isolated hardware, repeated trials) has been performed yet.

Software Testing Results

Real-Mode Pipeline Test

5 simulated patients successfully processed end-to-end through the live NATS pipeline:

PAT001 PAT002 PAT003 PAT004 PAT005

Each run produced: ECG prediction, X-ray status, overall severity, DB record, text report, CPU/RAM/process-time readings.

50-Patient Offline Simulation

50 patients processed and saved to the database in offline simulation mode. The pipeline generated results spanning all four severity levels:

NORMAL MODERATE HIGH CRITICAL

Offline Mode: Fault Tolerance

- If NATS is unavailable, the system automatically falls back to offline simulation mode.
- This allows AI inference, database logic, report generation, severity fusion, and performance monitoring to be tested without a running NATS server.
- **All results above are simulation / testing results, not real clinical patients.**

Raspberry Pi 5 + AI HAT+ Architecture (Planned)

Physical ECG Sensor → Raspberry Pi 5 → ECG Preprocessing → AI Inference → Hailo AI Acceleration → NATS / Local Processing → Database / Dashboard

Planned edge pipeline — sensor integration and Hailo acceleration are not yet implemented (see status slide).

Hardware Available

- Raspberry Pi 5 — 8 GB LPDDR4X RAM, quad-core Cortex-A76 CPU, 27 W USB-C supply, 128 GB microSD.
- Raspberry Pi AI HAT+ with Hailo AI acceleration chip.
- Wi-Fi connectivity and a laptop available for setup / remote access.
- Raspberry Pi OS installed; system successfully updated (apt update / upgrade). Confirmed: Linux raspberrypi, aarch64 kernel.

Not Yet Available

- HDMI monitor — not currently available.
- USB keyboard — not currently available.
- Physical ECG sensor — not yet connected.

Current Status: Completed vs. Remaining

COMPLETE

- ✓ ECG dataset integration
- ✓ ECG CNN training
- ✓ ECG test evaluation
- ✓ 5-class ECG model
- ✓ ECG inference integration
- ✓ NATS publisher / subscriber
- ✓ Multimodal fusion logic
- ✓ SQLite database
- ✓ Automated reports
- ✓ Streamlit dashboard
- ✓ Runtime monitoring
- ✓ Offline 50-patient simulation
- ✓ Raspberry Pi OS setup

IN PROGRESS / REMAINING

- o Real ECG sensor integration
- o Raspberry Pi software deployment
- o AI HAT+ integration
- o Hailo acceleration
- o Hardware end-to-end test
- o Improved ECG class-imbalance handling
- o Full CXR model training / validation
- o Clinical-grade validation

HONEST ASSESSMENT

Limitations and Future Work

Limitations

- ECG dataset is highly imbalanced.
- Rare heartbeat classes have lower recall.
- Current ECG model is not clinically validated.
- X-ray component is currently a prototype / software module.
- Current NATS sensor input is simulated.
- Real ECG sensor integration is pending.
- Raspberry Pi edge deployment is not yet fully completed.
- Hailo AI HAT acceleration has not yet been benchmarked.
- Current multimodal severity rules are prototype logic.
- Confidence values should not be interpreted as clinical certainty.

Future Work

- Class imbalance handling (weighted loss, oversampling, focal loss)
- ECG signal augmentation
- Improved CNN architecture
- Calibration of prediction confidence
- More robust cross-validation
- Explainable AI
- Real ECG sensor integration
- Real-time streaming
- Hailo acceleration
- Raspberry Pi deployment
- Better X-ray model
- Multimodal learned fusion
- Clinical validation
- Security and privacy
- Larger and more diverse datasets

CONCLUSION

An End-to-End AI-Assisted Multimodal Triage Prototype

- This integrates ECG AI, X-ray analysis, real-time NATS communication, triage fusion, a SQLite database, automatic reporting, and dashboard monitoring into one working pipeline.
- The ECG component was rebuilt on a proper MIT-BIH-derived train/test split and achieved 97.06% test accuracy, evaluated in detail at the per-class level.
- The project is presented honestly as a research prototype: class imbalance, chest X-ray model validation, real sensor integration, Raspberry Pi deployment, and hardware acceleration remain open work.

97.06%
ECG test accuracy

13 / 21
software milestones complete

8
hardware milestones remaining