iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** XI **Month of publication:** November 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.85298

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue XI Nov 2026- Available at www.ijraset.com*

# HeartGuard: An IoT-Enabled Cardiovascular Health Monitoring and Risk Prediction System

Dr. Kapil Gupta[1], Anchal Gupta[2], Shivam Kurhadkar[3], Palak Dugad[4], Junaid Hameed[5]
*Department of Computer Science and Engineering, St. Vincent Pallotti College of Engineering and Technology, Nagpur, India*

***Abstract:*** ***HeartGuard is a prototype system that uses Internet of Things (IoT) technology to monitor cardiovascular health and predict risks. It combines physiological sensing, ECG data capture, wireless communication, database storage, and patient medical profiles within one integrated platform. The hardware centers on an ESP32 microcontroller and includes sensors for heart rate, oxygen saturation ($SpO_2$), temperature, and ECG. The system displays data locally on an OLED screen and also sends it to a web application, allowing remote data viewing. The software integrates real-time sensor readings with patient details such as age, sex, blood pressure, cholesterol, chest-pain type, exercise-induced angina, ST depression, and resting ECG results. For risk prediction, HeartGuard uses a Random Forest machine learning model, while ECG arrhythmia classification relies on a convolutional neural network (CNN). Currently, these models are developed using publicly available data from Kaggle. In the future, HeartGuard aims to train models directly on data collected by the deployed sensors. The prototype presents cardiovascular risk as a percentage calculated at the application level, but future versions plan to use calibrated probabilities directly from the Random Forest output. The system includes interactive interfaces for risk visualization, ECG analysis, patient records management, and generates clinical-style PDF reports.***

***Keywords: Internet of Things; cardiovascular monitoring; ECG; physiological sensing; ESP32; machine learning; Random Forest; CNN; risk assessment; remote health monitoring.***

## I. INTRODUCTION

Monitoring heart health is increasingly supported by wearable and portable devices that collect physiological information outside traditional clinical environments. Continuous, long-term cardiac monitoring can complement periodic clinical tests, offering valuable insights between doctor visits. IoT technology extends this capability by connecting sensing devices to communication networks and software platforms, enabling remote storage and analysis of data.

HeartGuard was developed to bring these capabilities together in a single, healthcare-focused prototype. The system features an ESP32-based sensing layer, local OLED display, Wi-Fi communication, database-backed storage, patient medical profiles, ECG analysis, cardiovascular risk assessment, and report generation. It is designed as a research and decision-support tool rather than a clinically validated diagnostic device.

Key contributions of HeartGuard include:

- Integrating real-time physiological data collection with an IoT healthcare workflow
- Combining ECG acquisition and visualization with structured patient information
- Merging clinical profile data with IoT sensor measurements for cardiovascular risk assessment
- Implementing web dashboards to visualize risk and patient history
- Generating structured, clinical-style reports from collected data From a system design perspective, HeartGuard separates data acquisition, storage, inference, and presentation layers. This modularity allows independent inspection of sensor readings apart from risk dashboards and facilitates extending predictive models without redesigning the entire interface. It also clarifies which outputs come directly from hardware and which are generated by software.

## II. RELATED WORK

Remote cardiac monitoring has been studied extensively using wearable and portable devices because continuous data collection can provide valuable insights beyond what periodic clinical visits offer [1]. IoT-based healthcare systems further enhance this by connecting physiological sensors with remote storage and applications, allowing real-time monitoring and analysis from afar [2].

©IJRASET: All Rights are Reserved 

Various machine learning techniques have been applied to predict cardiovascular risk. These include Logistic Regression, Decision Trees, Random Forests, Support Vector Machines, K-Nearest Neighbors, Naive Bayes, gradient-boosting methods, and neural networks. Their effectiveness often depends heavily on the quality of datasets, preprocessing methods, feature representations, and validation protocols [4], [5], [15]. For instance, Predictis is a related IoT system that uses machine learning to classify cardiovascular risk levels [3].

Processing ECG data is another important research area. Traditional methods like the Pan–Tompkins algorithm provide a foundation for detecting QRS complexes and real-time ECG processing [6]. More recently, deep learning approaches, especially convolutional neural networks (CNNs), are increasingly used for classifying ECG signals. The MIT-BIH Arrhythmia database is a common benchmark dataset for developing and testing these methods [7], [8].

## III. PROBLEM STATEMENT AND OBJECTIVES

The project tackles the challenge of integrating multiple components—physiological sensing, wireless data transmission, storage, patient profile information, computational risk assessment, and user-friendly visualization—into a single, coherent workflow.

The practical goal is to move beyond isolated sensor measurements toward a platform where current physiological data can be combined with patient history and analyzed using machine learning models. This allows for a more comprehensive understanding of cardiovascular risk.

The implementation aims to:

- Acquire heart rate, $SpO_2$, temperature, and ECG signals
- Provide local visualization on an OLED display
- Transmit data over Wi-Fi and store timestamped readings
- Maintain detailed patient medical profiles
- Combine profile and sensor data for risk assessment
- Visualize ECG waveforms and maintain patient records
- Generate structured clinical-style reports

Operationally, the workflow is designed to support traceability. Every reading is linked to a timestamp and patient profile where applicable. The dashboard presents both raw sensor values and computed outputs together. The current implementation demonstrates these major stages, though a formal end-to-end reliability study is planned for future work.

## IV. SYSTEM ARCHITECTURE

The HeartGuard system is built in layers that move data from physical sensors all the way to user-friendly reports. At the base, physiological sensors measure heart rate, $SpO_2$, temperature, and ECG signals.

These readings are collected by an ESP32 IoT device, which performs data acquisition, shows information on a local OLED display, and sends the data via Wi-Fi.

The backend and database layer stores the measurements and makes them accessible to the HeartGuard application. This app combines medical profile information with the latest sensor readings and ECG data to perform predictions and generate reports.

A key design consideration is managing missing or delayed data. The application distinguishes between fresh readings and older stored values, showing timestamps to avoid confusion. If sensors are disconnected or the network fails, the interface prevents outdated data from being shown as if it were current. These behaviors are tested by deliberately disconnecting and reconnecting devices.

Conceptually, the data path can be seen as five logical layers:

*1)* Sensor interfaces that capture physiological signals
*2)* Embedded acquisition with OLED status display
*3)* Wireless data transfer via Wi-Fi
*4)* Backend/database services for storage
*5)* Web-based analysis and reporting

Separating these layers helps isolate issues, for example, distinguishing between a dashboard update failure and a local sensor problem [Fig. 1].

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

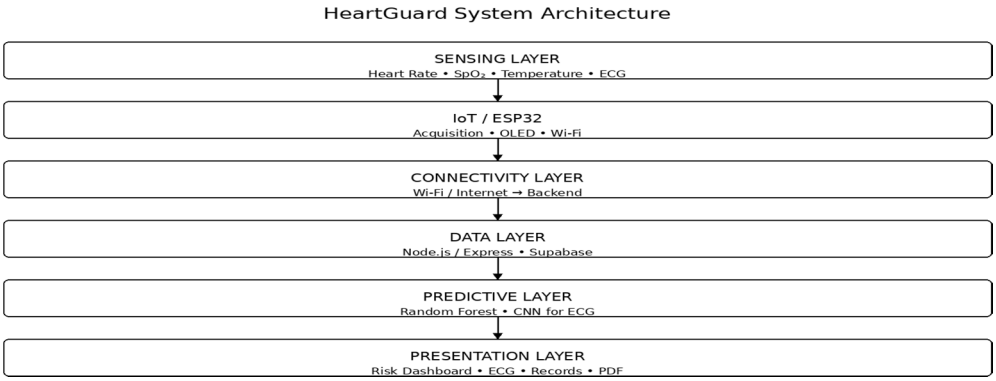



*Fig. 1. Overall HeartGuard architecture showing the sensing, IoT communication, data-storage, predictive, and presentation layers.*

## V. SYSTEM IMPLEMENTATION

### *A. Hardware Implementation*

The physical HeartGuard prototype is built on a breadboard. It includes an ESP32 controller, sensors for heart rate, $SpO_2$, temperature, ECG electrodes, an OLED display, and supporting electronic components. This hardware setup forms the critical interface between the sensors and the software platform.

### *B. Physiological Data Acquitization*

The system continuously measures heart rate, $SpO_2$, and temperature, while ECG signals are acquired through a dedicated path. The OLED display provides a real-time view showing heart rate (e.g., 79 BPM), $SpO_2$ (96%), temperature (32.3°C), and Wi-Fi status.
Because optical pulse sensors and contact-based ECG acquisition can be affected by motion, electrode contact quality, and electrical noise, future versions should include signal-quality indicators and detect missing or implausible data. The displayed values are prototypes and need validation against standard reference instruments under controlled testing protocols.

### *C. ECG Signal Acquitization and Processing*

The ECG module supports data loading, waveform visualization, heart rate display, arrhythmia classification, risk prediction, urgency indication, and clinical-style PDF report generation. For instance, a typical ECG recording is sampled at 360 Hz for about 10 seconds, showing a heart rate of 72 BPM with a “Normal” arrhythmia label.
The arrhythmia classification is done using a convolutional neural network (CNN), following approaches explored in recent ECG classification research [18].

### *D. IoT Communication*

The ESP32 serves as the communication bridge between the physical sensors and the software platform. The OLED display includes a Wi-Fi status icon, and the software sends collected health data to the backend and database layers. The prototype successfully demonstrates this communication path through interface updates and stored readings.

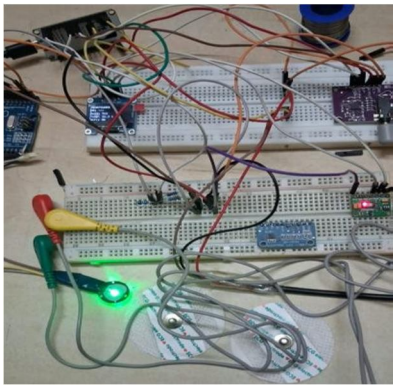


Fig. 2. Breadboard-based physical prototype of the HeartGuard IoT monitoring system.

Table I. Hardware Components And Functions

| Component | Function | Evidence |
|---|---|---|
| ESP32 / IoT controller | Sensor interfacing, processing, Wi-Fi | Prototype |
| Heart-rate sensor | Heart-rate acquisition | OLED + serial |
| $SpO_2$ sensor | Oxygen-saturation monitoring | OLED/dashboard |
| Temperature sensor | Temperature acquisition | OLED/dashboard |
| ECG sensing path | ECG acquisition | Electrodes + ECG UI |
| OLED display | Local visualization | Prototype |
| Breadboard wiring | Hardware integration | Prototype |

### *E. Data Storage and Database*

The software platform stores the latest and historical IoT sensor readings, timestamps, patient records, and risk summaries. It uses Supabase as the database backend, while Node.js and Express.js provide the server layer. This setup allows the application to retrieve stored measurements for combining with patient profiles, making predictions, visualizing dashboards, and generating reports. Data persistence is crucial because a single dashboard value doesn't provide historical context. Timestamped records enable comparison of readings over time and help link recent measurements with patient profile data. The current implementation does not quantify database latency, uptime, or throughput since those performance metrics have not been fully evaluated yet.

### *F. Medical Profile Integration*

The medical profile module stores baseline patient information, including:

- Age
- Biological sex
- Exercise-induced angina
- Blood pressure
- Cholesterol levels
- ST depression
- Chest-pain type
- Resting ECG result

During risk assessment, these clinical variables are combined with recent IoT sensor data. This integration allows physiological observations to be interpreted in the context of structured patient information, improving cardiovascular risk prediction.

## VI. MACHINE LEARNING METHODOLOGY

### *A. Dataset*

The current HeartGuard prototype uses a cardiovascular dataset from Kaggle for initial model development. This static dataset supports development while the project moves toward collecting a larger, labeled dataset from deployed HeartGuard sensors. The future dataset will pair sensor-derived physiological observations with appropriate clinical target labels.

A dataset shift is expected because models trained on public tabular data may not directly generalize to live sensor data. Differences in feature definitions, units, sampling windows, missing data handling, and target labels must be aligned before combining datasets. Sensor-derived training examples will require labeling through documented clinical protocols, as device measurements alone do not establish disease diagnoses.

### *B. Input Features*

The cardiovascular risk prediction model uses a combination of structured clinical and physiological data, including:

- Age
- Biological sex
- Chest-pain type
- Exercise-induced angina
- Blood pressure
- Cholesterol

- ST depression
- Resting ECG result
- Heart rate
- $SpO_2$ (oxygen saturation)
- Body temperature

These features correspond to the information provided by the medical-profile module and the real-time IoT sensor readings, enabling a comprehensive risk assessment that blends static clinical data with current physiological measurements.

## VII. MACHINE LEARNING-BASED RISK PREDICTION

HeartGuard implements the cardiovascular risk classifier using a Random Forest algorithm. Random Forest is an ensemble method that builds multiple decision trees trained with randomized subsets of data and features. This approach is well-suited for handling structured clinical and IoT variables and capturing non-linear relationships [9].

The model processes a combined feature vector made up of clinical and physiological variables. Preprocessing steps include categorical encoding (for variables like chest-pain type), normalization, and splitting data into training and test sets before model development. The same preprocessing workflow will be applied consistently when sensor-derived labeled data are incorporated into the training dataset.

It is important to explicitly define target variables. Public datasets often label patients by diagnosis or risk categories, which are not direct measurements from sensors. Heart rate, $SpO_2$, and temperature are observations but do not alone serve as ground-truth labels for cardiovascular disease. Therefore, labeled training data must clearly document the source and clinical meaning of each target.

For reproducibility, preprocessing parameters such as normalization scales should be learned from the training data only and applied unchanged to validation and test sets. Probability calibration should also be assessed separately from model discrimination performance [16]. Ideally, if multiple data points belong to the same patient, data splitting should be done at the patient level to avoid leakage between training and testing sets.

Currently, the HeartGuard prototype calculates cardiovascular risk as a percentage through an application-level process derived from the Random Forest output. Future versions intend to directly use calibrated probabilities from the model to present risk scores.

Selecting a decision threshold for categorizing risk is a design choice distinct from model training. Lower thresholds may catch more high- risk cases but increase false positives, while higher thresholds reduce false alarms but may miss some cases. Thresholds should be chosen based on validation data for the intended use case, and final testing should be done on untouched test sets. Model probabilities should not be taken as absolute clinical probabilities without proper calibration and validation. Evaluation metrics such as class balance, sensitivity, specificity, precision, recall, F1-score, ROC-AUC, and confusion matrices should be reported on independent test sets. For imbalanced datasets, precision-recall analysis complements ROC evaluation [17]. Additionally, it is crucial to consider the clinical context when interpreting model outputs. Continuous monitoring and periodic model retraining with new data will further enhance prediction accuracy and reliability over time.

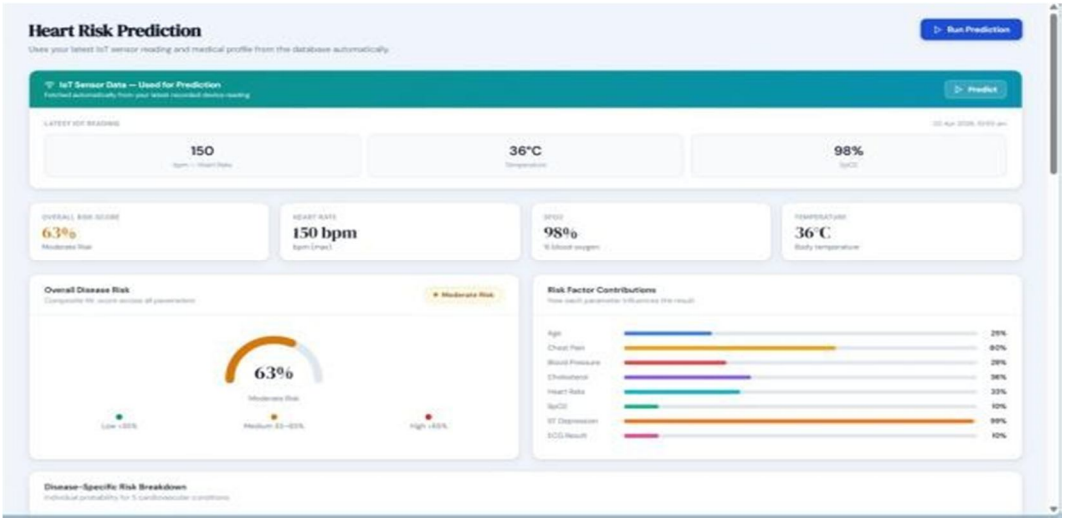



Fig. 4. HeartGuard risk-prediction dashboard integrating recent IoT readings with patient-profile information

### *A. Web Application Implementation*

The web application contains a heart-risk prediction dashboard, medical-profile interface, IoT sensor-data panel, ECG analysis module, and patient-record dashboard. The risk interface displays heart rate, $SpO_2$, temperature, overall risk score, risk category, factor contributions, and disease-specific breakdown. A demonstrated screen

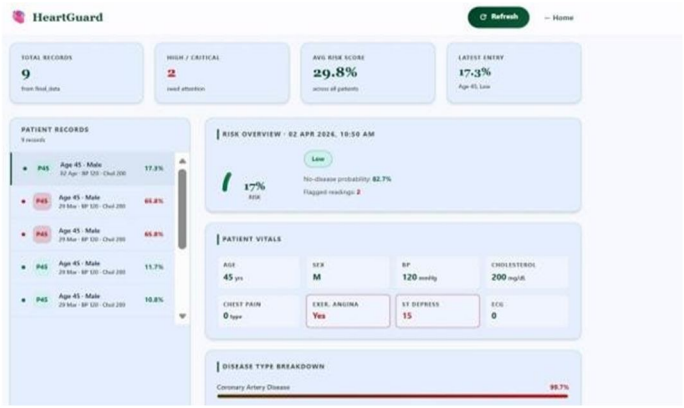



Fig. 5. Patient-record dashboard showing stored records, risk overview, patient vitals, and disease-specific breakdown.

### *B. Clinical Report Generation*

The ECG interface provides a clinical-style report workflow with ECG parameters, risk information, disease breakdown, interpretation, recommendations, and a doctor-stamp area, together with a PDF download function. The feature converts collected and analyzed information into a structured report for research and demonstration use.

The report interface is a presentation and documentation feature; it does not replace interpretation by a qualified clinician. In a future deployment, the report should identify the measurement time, data source, model/version, and limitations of the generated risk estimate so that users can distinguish measured values from algorithmic outputs

## VIII. EXPERIMENTAL SETUP AND RESULTS

The current evidence supports implementation-level evaluation. The physical prototype demonstrates physiological sensing, local OLED output, serial-monitor BPM readings, and web interfaces for stored and predicted information. These observations establish functional integration of the major system stages.

A useful next experiment is a staged integration test. First, record repeated sensor values and compare them with reference measurements. Second, verify that each reading reaches the database with the expected timestamp. Third, confirm that the dashboard retrieves the same stored values. Finally, test that profile changes affect the intended feature vector and that the generated report reflects the same inputs and model output. Logging failures at each stage would help distinguish hardware, communication, storage, and application errors. shows 150 BPM, 98% $SpO_2$, 36 °C, and a 63% moderate-risk score; these are interface observations rather than clinical validation results.

The evaluation in this paper is qualitative and based on the supplied prototype screenshots and implementation description. It documents visible functions and example values rather than a controlled clinical study. No numerical accuracy, sensitivity, specificity, latency, power-consumption, or patient-outcome claim is made because no reproducible measurement table for these metrics was supplied.

For sensor acquisition, the OLED output demonstrates HR = 79 BPM, $SpO_2$ = 96%, temperature = 32.3 °C, and Wi-Fi = OK. The serial monitor contains changing BPM observations followed by stable reported values, demonstrating the acquisition and stabilization workflow.

For ECG acquisition, the interface demonstrates a 360-Hz record containing 3600 samples over approximately 10 s with a displayed heart rate of 72 BPM. The ECG waveform is visualized through the web application and linked to analysis, risk, and report functions.

For end-to-end application behavior, the patient dashboard demonstrates stored records, risk summaries, patient vitals, flagged readings, and disease-type breakdown. Example values shown in the interface are treated as demonstration data and not as a clinical cohort result.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Table II. Software Technologies Used

| Layer | Technology | Function |
|---|---|---|
| Embedded | ESP32 | Sensor acquisition and Wi-Fi |
| Backend | Node.js, Express.js | Server/API layer |
| Database | Supabase | Cloud/database storage |
| Machine learning | Python, Scikit-learn | Random Forest risk model |
| ECG classification | CNN | Arrhythmia classification |

## IX. DISCUSSION

The implementation integrates physiological sensing, IoT communication, database storage, machine-learning analysis, ECG processing, dashboards, and reporting in one workflow. Research on structured electronic health records likewise highlights the value of carefully representing patient information for predictive modelling [19]. The primary contribution is system integration rather than a new machine-learning algorithm. The broader literature supports combining IoT/IoMT sensing with machine-learning analysis for cardiovascular monitoring [4].

The ECG module separates signal acquisition from interpretation: the waveform is acquired and visualized, while the CNN classification requires quantitative evaluation on labelled ECG data. The Random Forest stage currently uses development data and will transition toward a larger sensor-derived dataset. Deep-learning ECG classification has also been demonstrated on ambulatory ECG data [8], [9].

The current percentage is an application-level output. Future work will use calibrated Random Forest probability directly so that the meaning of the reported score can be evaluated quantitatively.

## X. LIMITATIONS AND FUTURE SCOPE

The present prototype is breadboard-based and intended for controlled development and demonstration. Sensor readings have not yet been systematically benchmarked against calibrated reference instruments. The cardiovascular model currently uses a Kaggle dataset, while future training will use labelled sensor-derived observations. Quantitative Random Forest and CNN evaluation remains to be completed.

Future work will focus on hardware miniaturization, sensor calibration, larger longitudinal data collection, direct Random Forest probability output, CNN evaluation, network-performance analysis, security/privacy, and external validation. ECG benchmark data may support reproducible evaluation, while device-derived data will capture real-world sensor noise and deployment conditions [7], [8].

## XI. VALIDATION, SAFETY, AND DEPLOYMENT CONSIDERATIONS

A reliable evaluation plan should test the hardware and software as separate components before assessing the complete workflow. Sensor measurements should be compared with suitable reference devices across repeated trials, with test conditions, sample counts, missing readings, and measurement error documented. Network tests should record successful and failed transfers and confirm that timestamps remain associated with the correct readings. These checks would make future results reproducible without treating a successful demonstration as evidence of clinical accuracy.

Model evaluation should use clearly defined target labels and a patient-independent test set. For the Random Forest component, a report should include the class distribution, confusion matrix, precision, recall, F1-score, sensitivity, specificity, ROC-AUC, and calibration analysis. For the CNN component, evaluation should describe the ECG dataset, sampling frequency, segmentation method, label mapping, patient-level split, and class-wise performance. Results should be compared with simple baselines and reported with enough detail to reproduce the experiment. No such quantitative performance values are asserted in this paper because they were not provided in the current project evidence.

Deployment also requires attention to consent and data minimization. Only fields necessary for the prototype should be collected, and development datasets should avoid unnecessary personal identifiers. Access to patient records and exported PDF reports should be restricted to authorized users. These are proposed safeguards for future deployment and should not be read as a claim that a formal security audit has already been completed.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Because HeartGuard handles health-related information, future deployment should use authenticated access, least-privilege database policies, protected transmission, careful handling of patient identifiers, and a documented retention policy. These safeguards align with broader guidance on privacy and security risks in connected healthcare systems [20]. The interface should clearly distinguish raw sensor readings, model outputs, and any application-level risk calculation. Users should be told that the current system is a prototype and that alerts or risk scores are not a diagnosis. Any real-world clinical use would require appropriate expert oversight, privacy review, safety testing, and validation in the intended population.

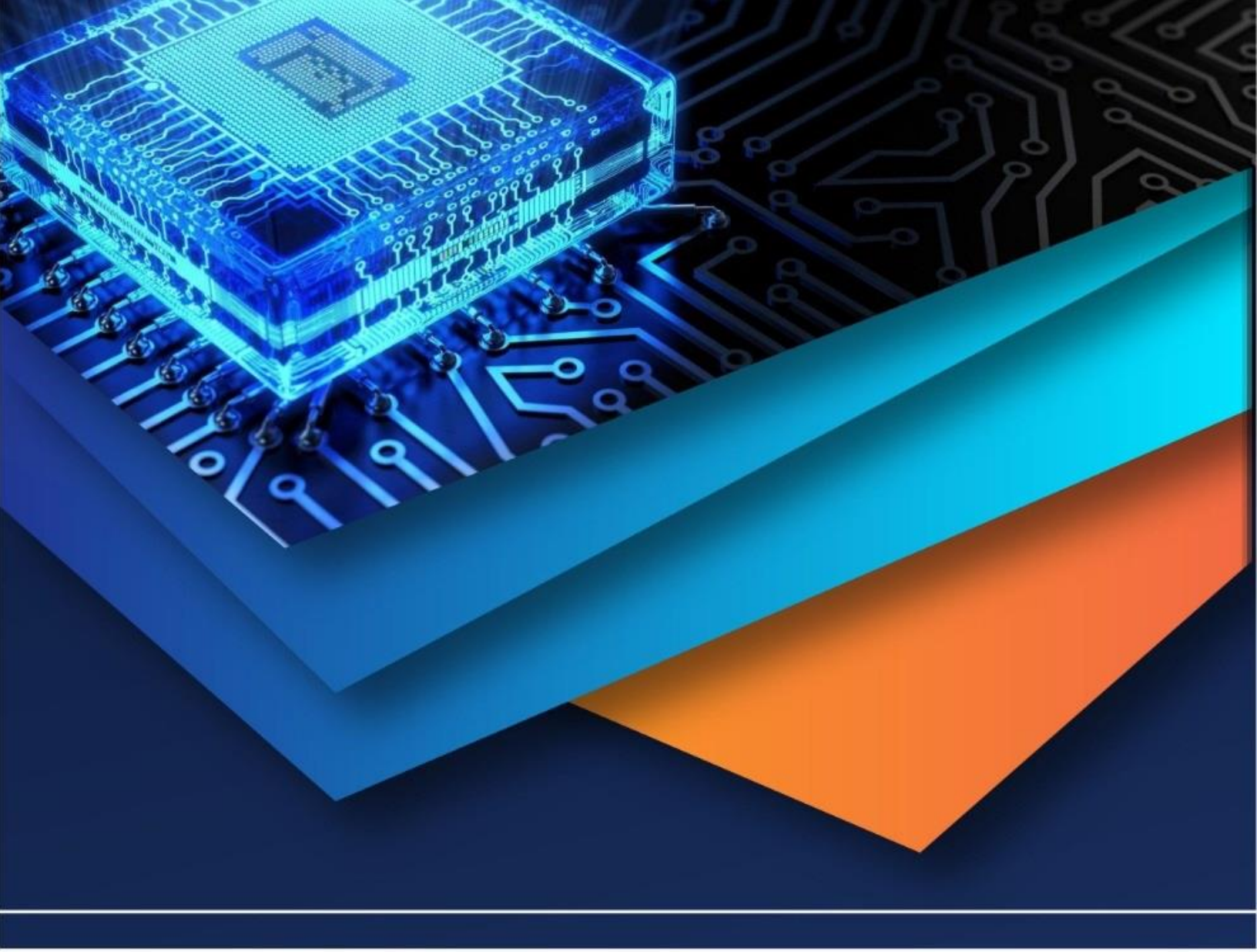
## XII. CONCLUSION

HeartGuard integrates IoT physiological monitoring, ECG acquisition, wireless communication, database storage, patient-profile management, Random Forest cardiovascular risk assessment, CNN-based ECG classification, web visualization, patient records, and report generation. The current Kaggle-based model provides a development foundation, while future labelled sensor measurements will support a system-specific dataset, calibrated risk probabilities, and more rigorous validation.

## REFERENCES

[1] Sana, E. M. Isselbacher, J. P. Singh, E. K. Heist, B. Pathik, and A. A. Armoundas, "Wearable Devices for Ambulatory Cardiac Monitoring: JACC State-of-the-Art Review," J. Amer. Coll. Cardiol., vol. 75, no. 13, pp. 1582–1592, 2020, doi: 10.1016/j.jacc.2020.01.046.
[2] H. Habibzadeh, K. Dinesh, O. R. Shishvan, A. Boggio-Dandry, G. Sharma, and T. Soyata, "A Survey of Healthcare Internet-of-Things (HIoT): A Clinical Perspective," IEEE Internet Things J., vol. 7, no. 1, pp. 53–71, 2020, doi: 10.1109/JIOT.2019.2946359.
[3] P. P. Ray, D. Dash, and N. Kumar, "Sensors for Internet of Medical Things: State-of-the-Art, Security and Privacy Issues, Challenges and Future Directions," Comput. Commun., vol. 160, pp. 111–131, 2020, doi: 10.1016/j.comcom.2020.05.029.
[4] A. Cuevas-Chávez, Y. Hernández, J. Ortiz-Hernandez, E. Sánchez-Jiménez, G. Ochoa-Ruiz, J. Pérez, and G. González-Serna, "A Systematic Review of Machine Learning and IoT Applied to the Prediction and Monitoring of Cardiovascular Diseases," Healthcare, vol. 11, no. 16, p. 2240, 2023, doi: 10.3390/healthcare11162240.
[5] G. Quer, R. Arnaout, M. Henne, and R. Arnaout, "Machine Learning and the Future of Cardiovascular Care: JACC State-of-the-Art Review," J. Amer. Coll. Cardiol., vol. 77, no. 3, pp. 300–313, 2021, doi: 10.1016/j.jacc.2020.11.030.
[6] J. Pan and W. J. Tompkins, "A Real-Time QRS Detection Algorithm," IEEE Trans. Biomed. Eng., vol. BME-32, no. 3, pp. 230–236, 1985, doi: 10.1109/TBME.1985.325532.
[7] G. B. Moody and R. G. Mark, "The Impact of the MIT-BIH Arrhythmia Database," IEEE Eng. Med. Biol. Mag., vol. 20, no. 3, pp. 45–50, 2001, doi: 10.1109/51.932724.
[8] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, A. Gertych, and R. S. Tan, "A Deep Convolutional Neural Network Model to Classify Heartbeats," Comput. Biol. Med., vol. 89, pp. 389–396, 2017, doi: 10.1016/j.compbiomed.2017.08.022.
[9] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng, "Cardiologist-Level Arrhythmia Detection and Classification in Ambulatory Electrocardiograms Using a Deep Neural Network," Nat. Med., vol. 25, no. 1, pp. 65–69, 2019, doi: 10.1038/s41591-018-0268-3.
[10] L. Breiman, "Random Forests," Mach. Learn., vol. 45, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
[11] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Proc. Advances in Neural Information Processing Systems 30 (NeurIPS), 2017, pp. 4765–4774.
[12] H. Xie, L. Yang, B. Jiang, Z. Huang, and Y. Lin, "State-of-the-Art Wearable Sensors for Cardiovascular Health: A Review," npj Cardiovascular Health, vol. 2, art. no. 53, 2025.
[13] S. E. El-deep, A. A. Abohany, K. M. Sallam, et al., "A Comprehensive Survey on Impact of Applying Various Technologies on the Internet of Medical Things," Artificial Intelligence Review, vol. 58, art. no. 86, 2025, doi: 10.1007/s10462-024-11063-z.
[14] P. Matthew, S. Mchale, X. Deng, G. Nakhla, M. Trovati, N. Nnamoko, E. Pereira, H. Zhang, and M. Raza, "A Review of the State of the Art for the Internet of Medical Things," Sci., vol. 7, no. 2, p. 36, 2025, doi: 10.3390/sci7020036.
[15] G. S. Collins, J. B. Reitsma, D. G. Altman, and K. G. M. Moons, "Transparent Reporting of a multivariable prediction model for Individual Prognosis Or Diagnosis (TRIPOD): The TRIPOD
[16] Statement," Ann. Intern. Med., vol. 162, no. 1, pp. 55–63, 2015, doi: 10.7326/M14-0697.
[17] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in Proc. 34th Int. Conf. Machine Learning (ICML), 2017, pp. 1321–1330.
[18] T. Saito and M. Rehmsmeier, "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets," PLoS ONE, vol. 10, no. 3, p. e0118432, 2015, doi: 10.1371/journal.pone.0118432.
[19] M. Kachuee, S. Fazeli, and M. Sarrafzadeh, "ECG Heartbeat Classification: A Deep Transferable Representation," in Proc. IEEE Int. Conf. Healthcare Informatics (ICHI), 2018, pp. 443–444, doi: 10.1109/ICHI.2018.00092.
[20] A. Rajkomar et al., "Scalable and Accurate Deep Learning with Electronic Health Records," npj Digit. Med., vol. 1, art. no. 18, 2018, doi: 10.1038/s41746-018-0029-1.
[21] H. Kruse, B. Frederick, T. Jacobson, and D. K. Monticone, "Cybersecurity in Healthcare: A Systematic Review of Modern Threats and Trends," Technol. Health Care, vol. 25, no. 1, pp. 1–10, 2017, doi: 10.3233/THC-161263.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)