Method and Application for Early Diagnostic and Prediction Based on Heart Rate Estimation Using Probability

Danish Sharok Alam Rojas , Leonardo Juan Ramirez Lopez * , Javier Rodriguez Velasquez

Posted Date: 15 April 2026

doi: 10.20944/preprints202604.1069.v1

Keywords: Holter-ECG; cardiac dynamics; probabilistic analysis; biomedical signal processing

 doi:10.20944/preprints202604.1069.v1

Article

Method and Application for Early Diagnostic and Prediction Based on Heart Rate Estimation Using Probability

Danish Sharok Alam Rojas [1], Leonardo Juan Ramirez Lopez [1,*] and Javier Rodriguez Velasquez [2]

[1] TIGUM Research Group of Universidad Militar Nueva Granada, Bogota, Colombia

[2] HARMONIK Research Group, Bogota, Colombia

* Correspondence: leonardo.ramirez@unimilitar.edu.co

Abstract

Long-term Holter analysis requires software tools capable of automating signal preprocessing, temporal segmentation, probabilistic computation, and result visualization in a reproducible and interpretable manner. In this research, a modular software system for automated analysis of cardiac dynamics was developed following a software engineering perspective and an iterative lifecycle based on Scrum, including requirements definition, sprint planning, development, integration, testing, review with a medical specialist, and refinement. The platform was designed to analyze standardized temporal windows of 12, 14, and 18 h extracted from original 24 h Holter-ECG recordings and integrates a frontend, a backend, and a Python® analytical engine within a unified client–server framework. It processes Excel or CSV files containing hourly average heart-rate values, performs structural validation, discretizes the data into 10 beats-per-minute intervals, constructs empirical probability distributions, identifies recurrent dynamic patterns, and generates structured JSON outputs for web-based visualization. A complementary preprocessing module was also implemented for raw PhysioNet ECG signal records, enabling the loading of .hea and .dat files, automated R-peak detection, and extraction of hourly average heart-rate values. The system was evaluated on 113 Holter records from three open-access databases: 85 from SHDB-AF, 19 from the Long-Term ST Database, and 9 from the MIT-BIH Normal Sinus Rhythm Database. Overall structural agreement at the record level was 58.4% (66/113). To conclude, this system provides a reproducible web application pipeline for Holter signal data processing and probabilistic cardiac dynamics analysis, integrating software development, preprocessing, classification, and interpretable visualization within a modular framework.

Keywords: Holter-ECG; cardiac dynamics; probabilistic analysis; biomedical signal processing

1. Introduction

Cardiovascular diseases remain one of the main causes of mortality worldwide [1]. In parallel with this clinical burden, the increasing volume and complexity of long-duration physiological recordings has created a growing need for software systems capable of processing biomedical signals in a structured, reproducible, and interpretable manner. In Holter-based cardiac assessment, the challenge is not limited to signal acquisition itself; it also includes data ingestion, preprocessing, temporal segmentation, analytical execution, visualization, and traceable output generation within a coherent computational workflow [2–4]. For this reason, software architecture has become a critical component of modern signal-analysis systems intended for real biomedical use.

In this context, Holter electrocardiography remains particularly relevant because it provides continuous cardiac monitoring over extended periods and allows the analysis of temporal variations that are often not captured by conventional short-duration ECG recordings [5–7]. However, the practical usefulness of Holter monitoring depends on more than the quality of the signal. It also

 doi:10.20944/preprints202604.1069.v1

depends on the availability of software tools capable of transforming raw or semi-structured data into analyzable representations while preserving consistency, transparency, and usability. In recent years, ECG-oriented software studies have highlighted the importance of complete processing pipelines and integrated analysis platforms for improving the handling, interpretation, and operational use of cardiac signals. For example, complete heart-rate extraction workflows and real-time ECG monitoring systems have shown how signal-processing logic can be translated into structured computational environments [25,26]. In many studies, the mathematical model is described in detail, but the software implementation is presented only as a secondary component, even though it is essential for reproducibility and operational deployment [2–4].

From a methodological perspective, cardiac dynamics have long been studied through nonlinear, fractal, and probabilistic approaches. Fractal descriptions of physiology, alterations associated with aging and disease, and correlation properties of RR interval dynamics have shown that cardiac behavior cannot always be adequately represented through isolated linear metrics alone [8–11]. In addition, probabilistic formulations have provided a way to organize hourly average heart-rate values into discrete ranges and compare their distributions across time, making it possible to distinguish characteristic dynamic patterns while preserving analytical transparency [12–15]. This feature is especially valuable for software development because transparent methods are generally easier to audit, reproduce, and integrate into modular computational environments than opaque black-box solutions.

In addition, the development of reproducible software pipelines in biomedical signal processing has also been strengthened by open-access data repositories. PhysioNet has become a central reference because it provides long-term ECG signals, metadata, and standard datasets that support independent validation and transparent comparisons between computational implementations [16]. More recently, databases such as SHDB-AF have expanded the possibilities for software validation on atrial fibrillation recordings, while the Long-Term ST Database and the MIT-BIH Normal Sinus Rhythm Database remain important resources for evaluating heterogeneous cardiac conditions and baseline physiological patterns [17–19]. These repositories are particularly useful in software-oriented studies because they allow the same computational framework to be tested on different data sources under standardized conditions.

Likewise, another important aspect of the current state of the art is the need to connect algorithmic performance with software usability and reproducibility. User-centered interface design, decision-support principles, and reproducibility-oriented development practices are now recognized as essential components of biomedical software systems [2–4]. This means that a useful analytical platform should not be restricted to classification accuracy alone. It should also provide modular separation of responsibilities, interoperable outputs, clear visualization, and a workflow that can be inspected and repeated by other researchers. In this context, previous software-based and mathematical developments, including probability-based and Zipf–Mandelbrot-based approaches, have shown that diagnostic support methods can be translated into computational tools with clinical interpretability [20,21].

At the theoretical level, the probabilistic perspective adopted in this field is also consistent with the broader foundations of probability theory and its interpretation in scientific reasoning [22–24]. Although historical foundations alone do not solve current implementation challenges, they reinforce the conceptual basis for representing cardiac behavior through distributions of occurrence rather than only through isolated measurements. When combined with biomedical software engineering principles, this perspective supports the design of systems that can convert hourly heart-rate values into structured probabilistic profiles, produce auditable outputs, and facilitate downstream visualization and interpretation.

Finally, within this context, the aim of the present work is to develop and describe a software system for probabilistic analysis of Holter-derived cardiac dynamics using standardized temporal windows of 12, 14, and 18 h extracted from full 24 h recordings. The proposed framework integrates raw ECG preprocessing, temporal segmentation, data validation, probabilistic classification, and

 doi:10.20944/preprints202604.1069.v1

web-based visualization into a single reproducible workflow. It combines a frontend for interaction, a backend for orchestration, and a Python® analytical engine for probabilistic processing, while also incorporating a preprocessing stage for raw PhysioNet .hea and .dat files [16–19]. Overall, this study shows that probabilistic cardiac dynamics analysis can be implemented as a modular, scalable, and interpretable software pipeline, where software architecture is not a secondary implementation detail but a central part of the scientific contribution [2–4,12,15,20,21].

2. Materials and Methods

A software-oriented system was developed for the probabilistic analysis of cardiac dynamics from long-term Holter-ECG records. The proposed framework integrated raw-signal preprocessing, temporal segmentation, probabilistic computation, and web-based visualization within a single modular workflow. The system was designed to receive either raw ECG records or structured hourly heart-rate files, transform them into standardized analytical inputs, and generate interpretable outputs through a client–server architecture. The following subsections describe the data sources, software components, preprocessing procedures, analytical methods, and validation strategy used in the implementation.

2.1. Data Sources

The software framework was evaluated using open-access long-term ECG records obtained from three PhysioNet databases. A total of 113 Holter recordings were included: 85 from SHDB-AF, 19 from the Long-Term ST Database, and 9 from the MIT-BIH Normal Sinus Rhythm Database [17–19]. SHDB-AF contains long-term Holter recordings from patients with atrial fibrillation and related rhythm changes, making it suitable for observing arrhythmic patterns over time [17]. The Long-Term ST Database includes prolonged ECG recordings with annotations related to ST-segment alterations and ischemia-associated episodes, which makes it useful for examining sustained pathological dynamics in long-term monitoring [18]. The MIT-BIH Normal Sinus Rhythm Database contains long-duration ECG recordings from subjects with clinically normal sinus rhythm and was used as a reference set for baseline physiological behavior [19]. From a software perspective, the use of these three datasets allowed the proposed framework to be tested on heterogeneous signal conditions while preserving a unified computational workflow.

2.2. Technology Selection through Decision Matrices

The software stack used in this study was selected through a structured decision-matrix approach. This methodology was applied to compare alternative technologies for the frontend, backend, and analytical engine using predefined technical criteria relevant to the development of a reproducible biomedical software system. The evaluated criteria included performance, ease of integration, implementation effort, maintainability, compatibility with the rest of the framework, and practical deployment considerations. Each candidate technology was scored on a 1–5 scale, and the final selection was based on the overall balance between functionality, modularity, and suitability for the proposed workflow. According to this evaluation, React was selected for the frontend, PHP for the backend, and Python® for the analytical module.

The frontend decision matrix was designed to compare candidate technologies for implementing the web interface of the platform. The evaluation considered graphics rendering, complex state management, development time, bundle size, and ecosystem maturity. These criteria were chosen because the frontend was expected to support file upload, execution feedback, dynamic views, and interactive visualization of analytical outputs.

React achieved the highest overall score because it offered the best balance between performance, flexibility, and component-based development. Its architecture is well suited to reactive interfaces in which multiple views depend on a shared analytical state, such as upload panels, progress indicators, diagnostic cards, charts, and data tables. In addition, its mature ecosystem and

 doi:10.20944/preprints202604.1069.v1

compatibility with data visualization libraries made it especially suitable for a software environment requiring modularity and maintainability. The frontend evaluation results are summarized in Table 1.

Table 1. Frontend decision.

Criterion	React	Vue	Angular	Vanilla JS®
Graphics rendering	5	4	3	2
Complex state management	5	4	5	1
Development time	4	5	3	2
Bundle size	3	4	2	5
Ecosystem maturity	5	4	4	1
TOTAL	22	21	17	11

The backend decision matrix was used to compare alternative technologies for handling file processing, request orchestration, communication with the analytical engine, and structured response delivery to the frontend. The evaluated criteria were legacy compatibility, prototyping speed, memory efficiency, database connectivity, and cloud deployment. These criteria were selected because the backend in the proposed framework was intended to function as a lightweight but reliable orchestration layer between the user interface and the Python® processing module.

PHP achieved the highest overall score. Although Node.js and Python® offered strong prototyping capabilities, PHP provided the most balanced combination of compatibility, deployment simplicity, and practical suitability for a request-driven workflow based on file upload, server-side validation, external script execution, and JSON response handling. Its mature ecosystem and straightforward integration with Apache-based environments also made it appropriate for the implementation of a modular client–server architecture. The results of the backend evaluation are summarized in Table 2.

Table 2. Backend decision.

Criterion	PHP	Node.js	Flask	.NET
Legacy compatibility	5	3	2	4
Prototyping speed	3	5	5	2
Memory efficiency	4	3	3	2
Database connectivity	5	4	4	5
Cloud deployment	4	5	5	3
TOTAL	21	20	19	16

The analytical engine decision matrix was used to compare programming environments for implementing the preprocessing and probabilistic analysis modules of the system. The evaluated criteria were processing speed, backend compatibility, memory usage, statistical functions, and license cost. These criteria were selected because the analytical component was required to process ECG-derived data, compute probabilistic distributions, and integrate efficiently with the rest of the software architecture.

Python® achieved the highest overall score. Although C++ offered advantages in raw processing speed and memory efficiency, Python® provided a more balanced environment for the proposed application due to its strong compatibility with backend execution, broad availability of scientific libraries, and ease of development. Compared with R® and MATLAB®, Python® also offered a more practical combination of analytical capability, integration flexibility, and accessibility. In particular, libraries such as Pandas and NumPy made Python® especially suitable for preprocessing structured physiological data and implementing the probabilistic workflow of the system. The results of the analytical engine evaluation are summarized in Table 3.

 doi:10.20944/preprints202604.1069.v1

Table 3. Analytical language decision

Criterion	Python®	C++	R®	MATLAB®
Processing speed	4	5	2	4
Backend compatibility	4	3	2	4
Memory usage	4	5	2	4
Statistical functions	5	5	5	1
License cost	5	5	5	1
TOTAL	22	20	16	18

2.3. Software Architecture

The system was developed as a modular software architecture designed to separate data ingestion, preprocessing, analytical execution, and result visualization into interoperable components. This organization was intended to support reproducible processing of long-term Holter-ECG data while maintaining a clear distinction between user interaction, workflow orchestration, and probabilistic computation. By distributing these functions across dedicated modules, the framework facilitates maintainability, scalability, and the integration of both raw ECG records and structured input files within a unified computational environment.

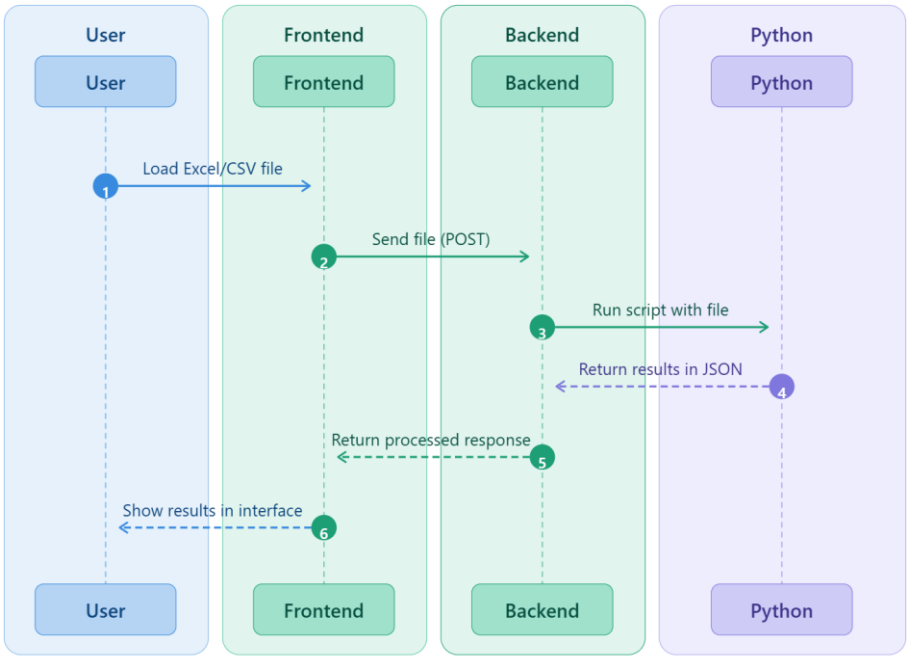


Figure 1. Sequence diagram of the software workflow for Holter-ECG analysis.

Figure 1 shows the sequential interaction among the four main components of the proposed system: the user, the frontend, the backend, and the Python® analytical engine. The workflow begins when the user loads an Excel or CSV file containing the input data through the frontend interface. The frontend then sends the file to the backend through an HTTP POST request. Once received, the backend calls the Python® module, which processes the file and executes the probabilistic analysis. The analytical results are returned from Python® to the backend in JSON format, and the backend subsequently sends the processed response to the frontend. Finally, the frontend displays the results in the user interface. This sequence reflects the modular client–server organization of the system and shows how data move from user input to analytical output in a structured and reproducible manner.

 doi:10.20944/preprints202604.1069.v1

With this general workflow established, the architecture of each software section can now be described in greater detail, starting with the frontend and its internal component organization.

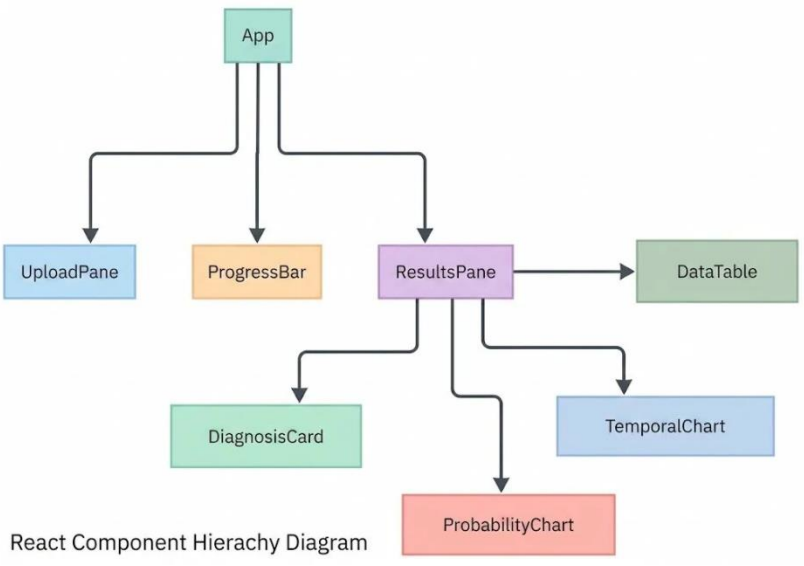


Figure 2. React component hierarchy of the web interface.

Figure 2 shows the component hierarchy used to implement the web interface in React. The application was structured as a set of reactive and reusable components organized around the main workflow of the platform. At the top level, the App component coordinates the three primary stages of user interaction: file upload, processing status, and result presentation. The UploadPane component manages file selection and drag-and-drop input for Excel or CSV files, the ProgressBar component provides visual feedback during execution, and the ResultsPane component organizes the analytical outputs. Within the results section, specialized child components were used to present different types of information, including DiagnosisCard for the main classification output, TemporalChart for time-based heart-rate visualization, ProbabilityChart for the probabilistic distribution across intervals, and DataTable for the structured tabular view of the processed values.

The interface was developed in React under the supervision of Dr. Javier Rodríguez Velázquez, with emphasis on a clinically intuitive workflow and clear visual hierarchy. The use of a component-based architecture made it possible to isolate responsibilities, simplify maintenance, and ensure that each part of the interface responded consistently to the analytical state of the application. Only functional interface elements were retained, prioritizing clarity, usability, and efficient interpretation of the results within a minimal and clinically oriented design.

 doi:10.20944/preprints202604.1069.v1

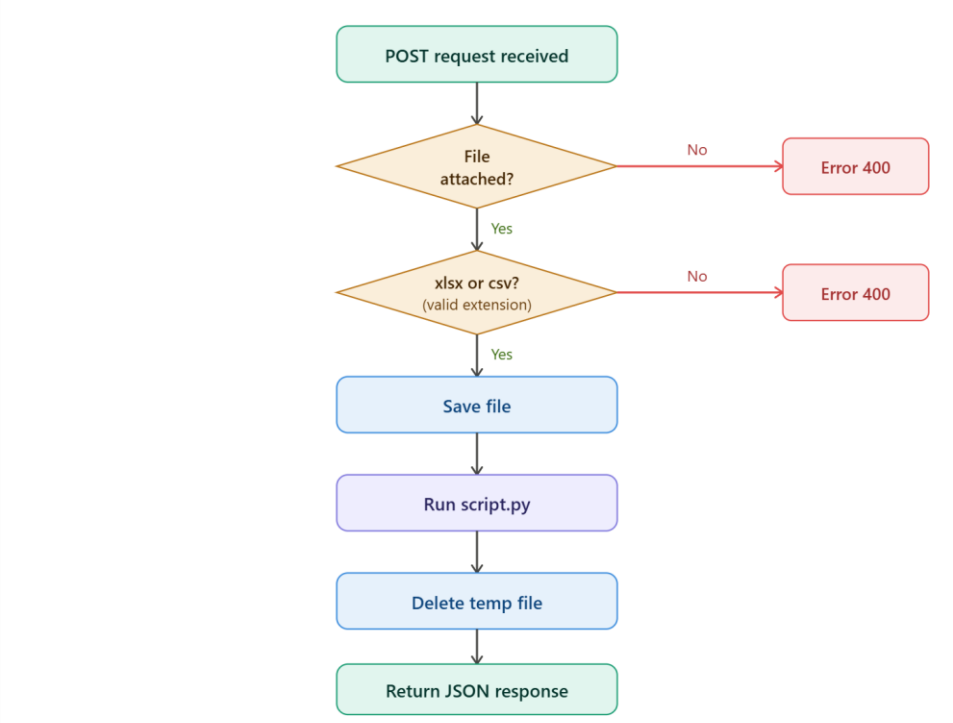


Figure 3. Simplified backend flow for file validation and analysis execution.

Figure 3 summarizes the backend workflow used to process uploaded files before analytical execution. Once a POST request is received, the backend first verifies whether a file is attached. If no file is detected, the system returns an error response. When a file is present, the backend validates whether the extension corresponds to one of the accepted input formats (.xlsx or .csv). If the format is invalid, an error response is generated. Otherwise, the file is temporarily stored on the server, passed to the Python® processing script, and removed after execution. Finally, the backend returns the analytical output to the frontend in JSON format. This flow reflects the validation and request-handling logic of the PHP backend and shows the minimal sequence required to connect file upload with probabilistic analysis.

 doi:10.20944/preprints202604.1069.v1

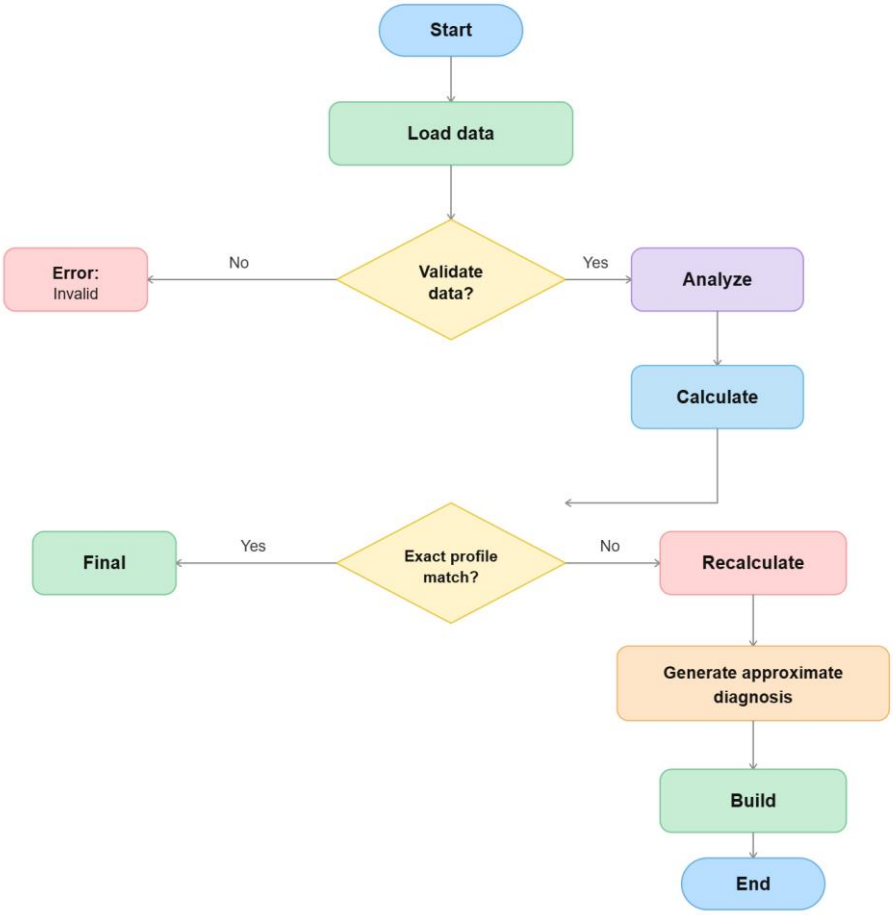


Figure 4. Flowchart of the probabilistic analysis algorithm.

Figure 4 presents the workflow of the analytical algorithm implemented in the Python® module. The process begins with data loading, followed by an initial validation stage to verify that the input is suitable for analysis. If the data do not meet the required conditions, the algorithm stops and returns an invalid-data error. When the input passes validation, the system proceeds to the analysis and calculation stages, where the probabilistic profile is computed from the processed values. The resulting profile is then compared against the expected reference patterns. If an exact profile match is found, the process reaches the final classification directly. Otherwise, the algorithm performs a recalculation step and generates an approximate diagnosis before building the final structured output. This flowchart summarizes the internal decision logic of the analytical engine and shows how the system handles both exact and approximate classification paths in a controlled and interpretable manner.

2.4. Software Development

The software was developed through an iterative lifecycle inspired by Scrum, as shown in Figure 5. The process began with requirements definition and sprint planning, followed by development, integration, and testing stages. Each cycle included review sessions with the medical specialist, allowing functional and usability adjustments to be incorporated through refinement. This iterative strategy supported the gradual consolidation of the system as a coherent software framework for Holter-ECG analysis.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 April 2026 doi:10.20944/preprints202604.1069.v1

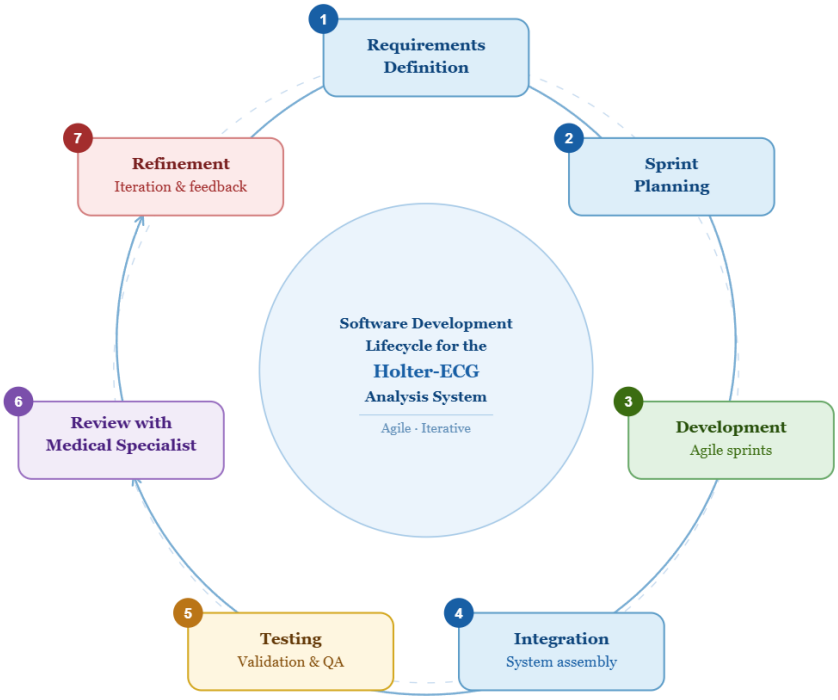
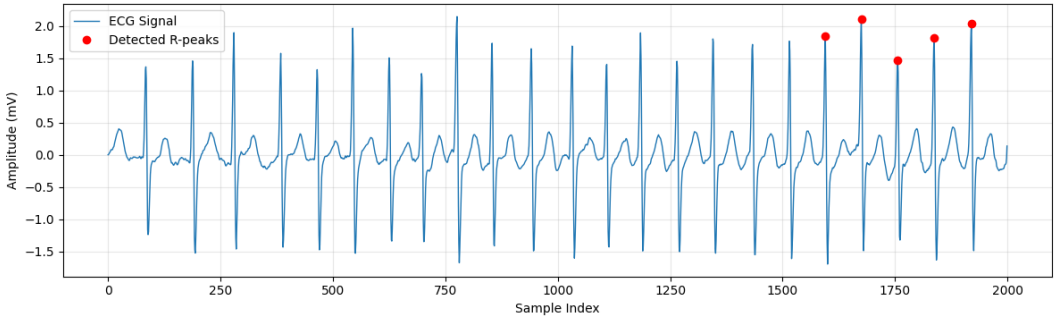


Figure 5. Software development lifecycle of the proposed Holter-ECG analysis system.

Figure 5 illustrates the software development lifecycle followed in the implementation of the proposed system. The process was organized as an iterative and agile cycle composed of seven main stages: requirements definition, sprint planning, development, integration, testing, review with the medical specialist, and refinement. This structure reflects the Scrum-based approach adopted during the project, in which each cycle was used to progressively improve the preprocessing modules, backend integration, analytical functions, and frontend components. The inclusion of a dedicated review stage with the medical specialist was essential to verify the coherence of the workflow, the interpretability of the outputs, and the usability of the interface. The refinement stage incorporated the feedback obtained in each iteration, allowing the platform to evolve as an integrated and reproducible software product.

2.5. Raw ECG Signal Preprocessing

A dedicated preprocessing module was developed to transform raw long-term ECG records from PhysioNet into structured inputs suitable for probabilistic analysis. This stage was designed to bridge the gap between original signal recordings and the analytical workflow of the software by converting raw .hea and .dat files into hourly average heart-rate values. The preprocessing routine included signal loading, R-peak detection, beat-to-beat interval extraction, heart-rate computation, and hourly aggregation. By structuring the signal in this way, the system generated standardized inputs that could be consistently used across different temporal windows and datasets within the same software environment.



 doi:10.20944/preprints202604.1069.v1

Figure 6. Initial ECG segment with detected R peaks during the preprocessing stage.

Figure 6 shows an initial segment of the ECG signal after loading and preprocessing, together with the R peaks detected by the automated routine. The blue trace represents the raw ECG waveform, while the red markers indicate the detected R-peak positions used to derive beat-to-beat temporal information. This step is essential because it provides the basis for RR interval calculation and subsequent heart-rate estimation. In the proposed software pipeline, accurate R-peak identification is the first key operation required to convert raw long-term ECG recordings into structured hourly average heart-rate values for downstream probabilistic analysis.

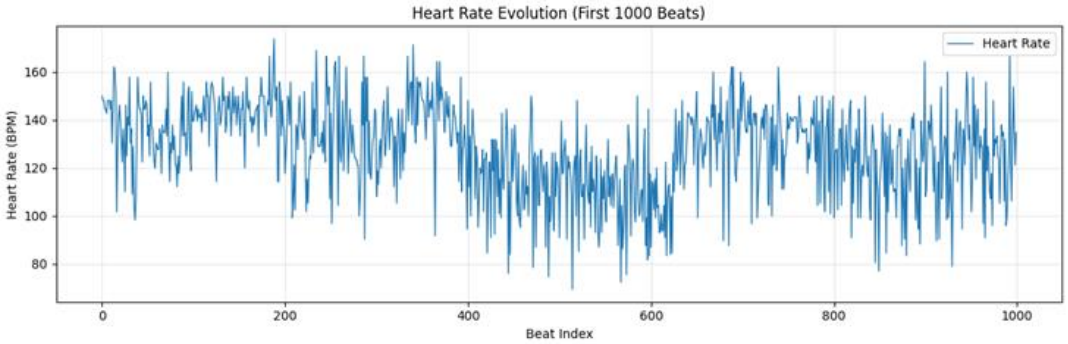


Figure 7. Beat-to-beat heart-rate evolution during the first 1000 detected beats.

Figure 7 shows the heart-rate evolution calculated from the first 1000 detected beats after R-peak identification. Each point in the series represents the instantaneous heart rate derived from successive beat intervals, allowing short-term fluctuations in cardiac dynamics to be observed before hourly aggregation. This intermediate step is important in the preprocessing pipeline because it verifies that the signal contains sufficient temporal resolution and variability for subsequent transformation into structured hourly average heart-rate values. In the proposed software framework, this stage serves as a bridge between raw ECG peak detection and the generation of standardized inputs for probabilistic analysis.

Table 4. Hourly average heart rate values obtained from a 24 h Holter-ECG record.

Hour	Average Heart Rate (BPM)	Hour	Average Heart Rate (BPM)
0	117.57	12	54.3
1	75.26	13	57.81
2	62.61	14	54.83
3	62.49	15	52.07
4	64.93	16	53.33
5	65.13	17	56.07
6	55.85	18	50.98
7	61.23	19	51.19
8	61.66	20	56.48
9	54.41	21	54.53
10	56.14	22	61.87
11	60.32	23	75.84

In Table 4, the hourly average heart-rate values extracted from a 24 h Holter-ECG record from the SHDB-AF database are presented. Each value corresponds to the mean heart rate calculated for one hour of the recording, generating a structured temporal profile of cardiac behavior across the full monitoring period. This hourly representation was used as the standardized input for subsequent temporal window selection and probabilistic analysis within the proposed software framework.

 doi:10.20944/preprints202604.1069.v1

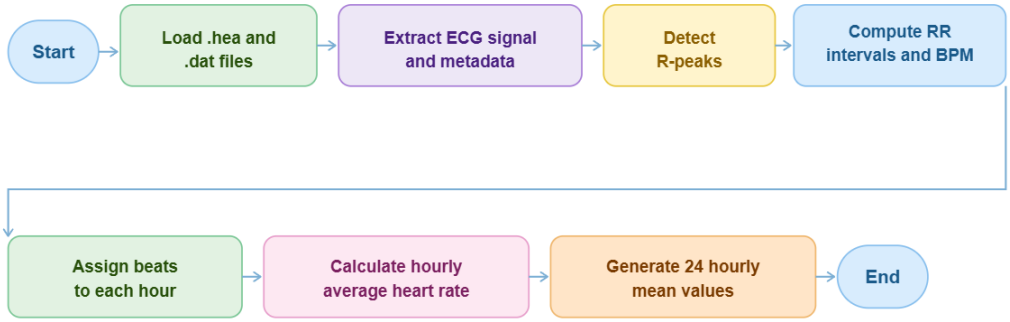


Figure 8. Preprocessing workflow for generating average heart-rate values from ECG records

Figure 8 shows the preprocessing workflow implemented in the software to transform raw PhysioNet Holter-ECG records into structured hourly average heart-rate values for downstream probabilistic analysis. The process begins with the loading of the .hea and .dat files, from which the system extracts both the ECG signal and its associated metadata. The preprocessing module then performs automated R-peak detection, computes RR intervals, and converts them into beat-per-minute values. After this step, the detected beats are assigned to their corresponding hourly segments, allowing the software to calculate the average heart rate for each hour of the recording. The final output of this pipeline is a structured 24 h heart-rate profile that serves as standardized input for the analytical modules of the proposed framework.

2.6. Probabilistic Analysis Framework

The probabilistic component of the software was based on the analysis of hourly average heart-rate values derived from long-term Holter-ECG recordings. Although the original signals corresponded to 24 h monitoring, the analytical workflow was designed to operate on standardized temporal windows of 12, 14, and 18 h in order to preserve comparability across records. For each selected window, the hourly averages were discretized into 10 beats-per-minute intervals within the 41–170 bpm range, allowing the construction of a discrete empirical probability distribution from the relative frequency of occurrence of each interval. Instead of relying on rigid predefined diagnostic labels, the system identified recurrent probabilistic distribution patterns and organized them into representative profiles, later interpreted as broad dynamic behaviors. This approach allowed the analytical logic of the software to remain data-driven while preserving interpretability and reproducibility.

3. Results

The purpose of the proposed software was to provide a structured and reproducible environment for processing Holter-ECG data, from data ingestion to probabilistic output visualization. In this context, the results presented in this section focus on two complementary aspects of the system. First, the performance of the web platform is described in terms of file upload, execution flow, and presentation of analytical outputs through the user interface. Second, the system is evaluated on open-access PhysioNet databases to examine how the software responds to heterogeneous long-term ECG records within a unified computational framework.

3.1. Web Interface and Execution Results

The software provided a web-based environment for uploading hourly average heart-rate data, executing the probabilistic analysis, and presenting the results in a structured and interpretable manner. In this context, the web interface acted as the visible execution layer of the system, connecting user input with the backend and the analytical engine. The results presented in this

 doi:10.20944/preprints202604.1069.v1

subsection describe how the platform supported file upload, execution monitoring, and dynamic visualization of the outputs generated by the software.

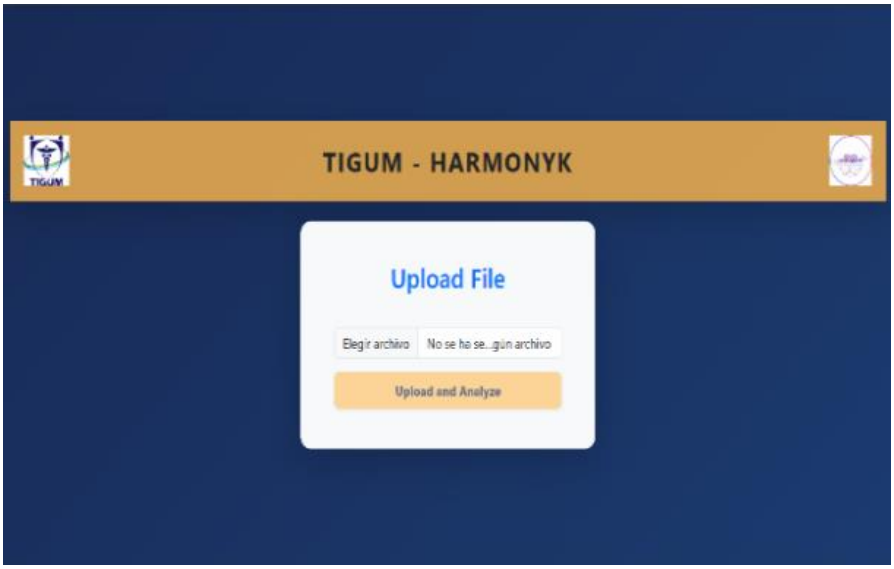


Figure 9. Initial web interface for uploading hourly average heart-rate files.

Figure 9 shows the initial web interface of the platform, where the user or the medical specialist can upload the hourly average heart-rate values in Excel or CSV format to begin the analysis. The upload module was designed as the entry point of the software workflow, providing a simple and centralized environment for file selection and execution. From this stage, the data are transferred to the backend and analytical modules, initiating the probabilistic processing pipeline within the web-based system.

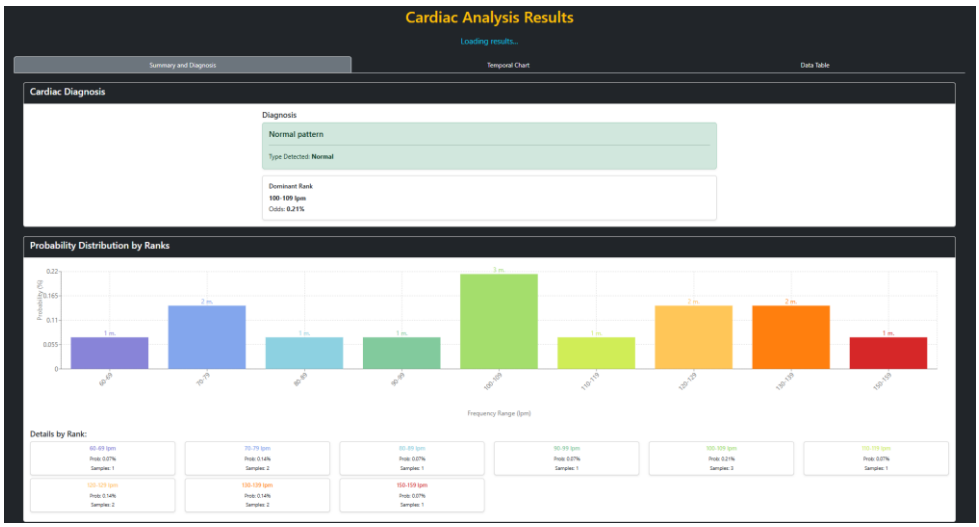


Figure 10. Web-based results interface for probabilistic cardiac dynamics analysis.

Figure 10 shows the results interface generated after execution of the probabilistic analysis. In this view, the software presents the main diagnostic output, including the identified cardiac pattern and the dominant probabilistic range, together with a graphical representation of the probability distribution across heart-rate intervals. Additional structured details are displayed below to support interpretation of the analytical result. The final stage of the execution workflow, where the processed data are transformed into interpretable outputs within the web-based environment.

 doi:10.20944/preprints202604.1069.v1

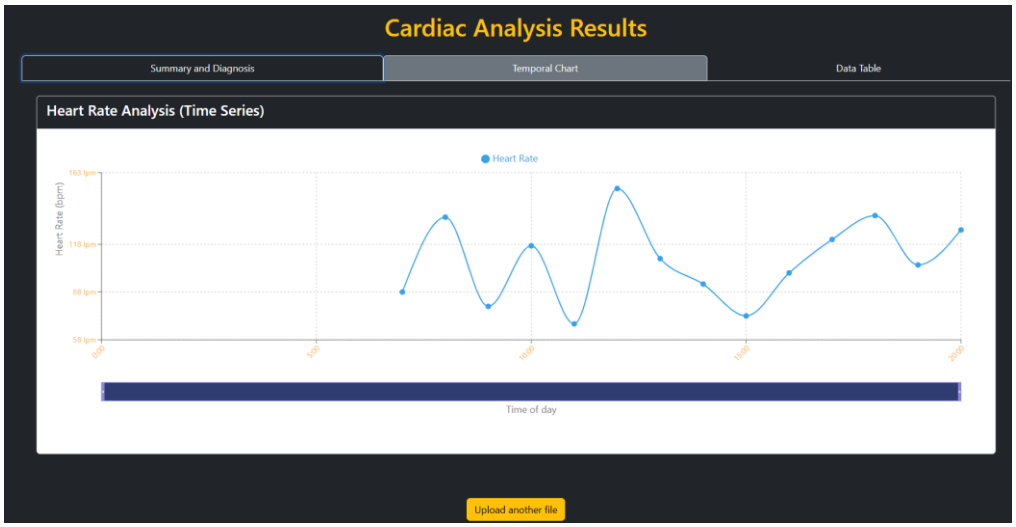


Figure 11. average hourly frequency over time.

Figure 11 shows the temporal chart section of the results interface after completion of the probabilistic analysis. In this view, the software displays the time-series behavior of the hourly average heart-rate values across the selected analytical window, allowing the user to observe fluctuations in cardiac activity over time. This graphical representation complements the diagnostic and probabilistic outputs by providing a direct view of the temporal structure of the processed data. Within the web platform, this component forms part of the integrated results environment, where summary diagnosis, temporal evolution, and tabular outputs are presented in coordinated views for interpretation.

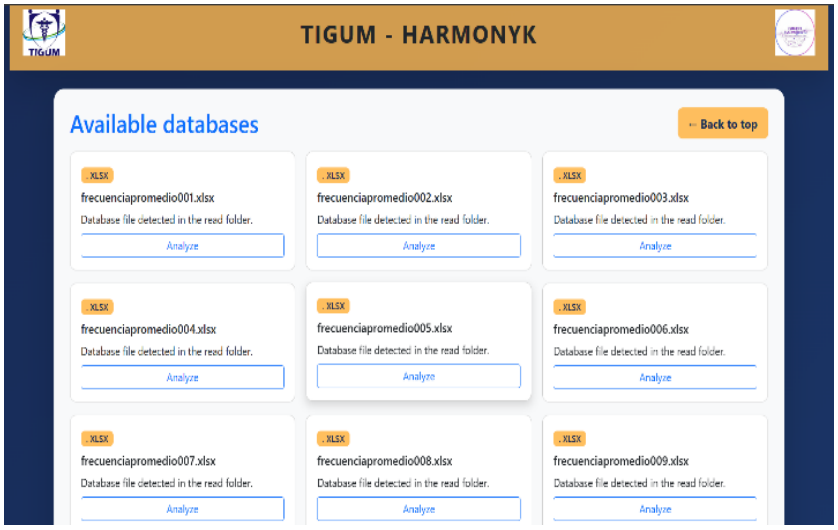


Figure 12. Web interface for browsing and selecting available database-derived records.

Figure 12 shows the database browsing interface of the web platform, where the records available for analysis are displayed as individual file cards. Each card corresponds to a structured Excel file generated from the preprocessing of long-term ECG recordings and includes an action button to launch the analysis directly from the interface. This view allows the user to navigate the available database-derived inputs in an organized way and select a specific record for execution within the same software environment.

 doi:10.20944/preprints202604.1069.v1

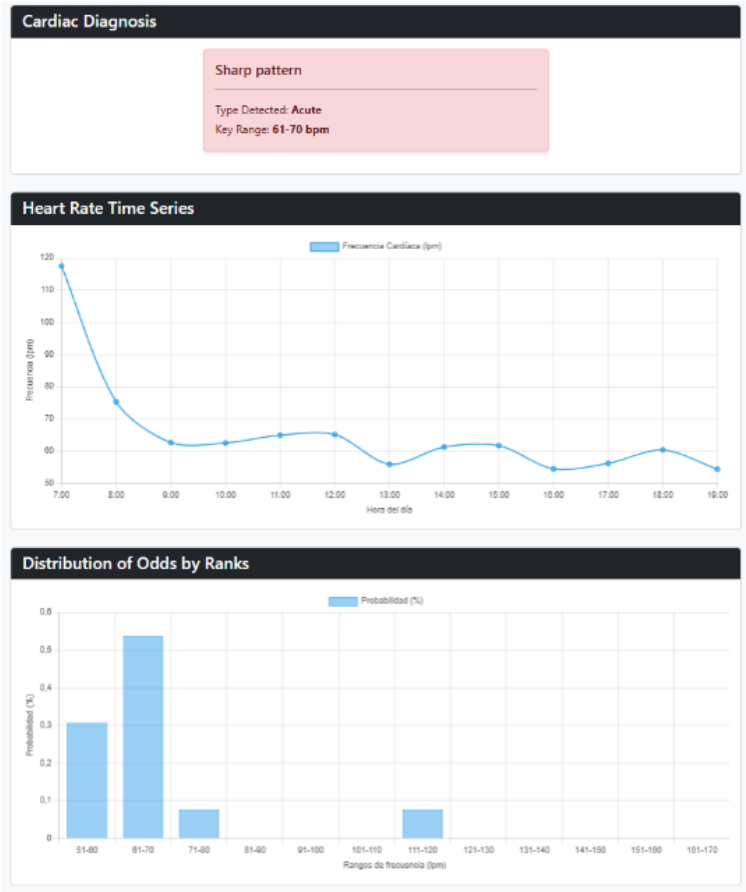


Figure 13. Integrated results showing diagnostic output, temporal behavior, and probabilistic distribution.

Figure 13 shows an integrated results view generated by the software after completion of the probabilistic analysis. In this interface, the system presents the detected cardiac pattern together with the assigned dynamic class and the key probabilistic range. The results are complemented by a temporal chart displaying the evolution of the hourly average heart-rate values and by a bar chart summarizing the probability distribution across the defined heart-rate intervals. This combined layout allows the user to examine the diagnostic output, the temporal structure of the processed data, and the interval-based probabilistic behavior within a single coordinated view.

3.2. Database-Based Validation Results

The proposed framework was applied to 113 long-term Holter-ECG records obtained from three PhysioNet databases: 85 from SHDB-AF, 19 from the Long-Term ST Database, and 9 from the MIT-BIH Normal Sinus Rhythm Database. The same computational workflow was applied across all records, including preprocessing when required, extraction of hourly average heart-rate values, construction of probabilistic distributions, and classification of the resulting patterns.

In the MIT-BIH Normal Sinus Rhythm Database, the software produced a balanced distribution of outputs, with three records classified as normal, three as chronic, and three as acute. In SHDB-AF, the predominant pattern identified by the system was acute, with 51 records classified as acute and 34 as chronic. In the Long-Term ST Database, the software predominantly identified chronic patterns, with 12 records classified as chronic and 7 as acute.

When these outputs were compared with the predominant expected condition of each database, the overall structural agreement at record level was 58.4% (66/113). This agreement should be interpreted as consistency between the probabilistic structures identified by the software and the general clinical characterization of each dataset, rather than as beat-level diagnostic accuracy. In this sense, SHDB-AF showed a predominant acute structural behavior, the Long-Term ST Database showed a predominant chronic structural behavior, and the MIT-BIH Normal Sinus Rhythm

 doi:10.20944/preprints202604.1069.v1

Database produced a uniform distribution across the three classes (3 records each), which is consistent with the limited sample size of this dataset (n = 9) and suggests that the probabilistic profiles derived from a small number of records may not converge toward a dominant pattern. This behavior should be interpreted as a consequence of sample insufficiency rather than a classification failure.

4. Discussion

The main contribution of the proposed work lies in translating probabilistic cardiac dynamics analysis into a structured software workflow rather than presenting it only as a mathematical procedure. From a software engineering perspective, the system transforms a process that would otherwise require several manual steps into a reproducible, deterministic, and standardized computational pipeline. This is especially relevant in Holter-based analysis, where conventional interpretation often depends on visual inspection and isolated heart-rate metrics. In this respect, the present work is consistent with previous ECG-oriented software studies that have emphasized the importance of complete processing pipelines and integrated computational environments for handling cardiac signals in a reproducible way [25,26]. However, unlike approaches mainly focused on signal acquisition, real-time monitoring, or heart-rate extraction, the proposed framework integrates preprocessing, probabilistic transformation, classification, and web-based visualization within a single software environment. By organizing the workflow into a unified execution platform, the system reduces operational variability, supports traceable processing, and enables the generation of interpretable outputs under controlled conditions.

A second important aspect is the modular structure of the platform. The separation between frontend, backend, and the Python analytical engine improves maintainability and scalability by allowing each layer to operate independently while remaining integrated within the same framework. The frontend manages user interaction and result visualization, the backend controls file validation and execution requests, and the analytical module performs preprocessing and probabilistic computation. In addition, the system supports both structured Excel/CSV inputs and raw PhysioNet .hea and .dat records, which broadens its applicability beyond a single input format. This dual-path design, together with the use of structured JSON outputs, strengthens interoperability and makes the platform more suitable for integration with other analytical tools or digital health environments. This interpretation is also aligned with previous work on biomedical software systems, where usability, modularity, and reproducibility have been identified as essential requirements for practical deployment beyond isolated analytical scripts [2–4].

The present study also has limitations that should be acknowledged. The overall structural agreement of 58.4% reflects dataset-level consistency between probabilistic profiles and broad clinical characterizations, and should not be interpreted as diagnostic accuracy in a clinical sense. Furthermore, the MIT-BIH Normal Sinus Rhythm Database, with only 9 records, did not provide sufficient statistical support to identify a dominant probabilistic pattern, and the resulting uniform distribution across classes should be interpreted as a consequence of sample insufficiency rather than a classification failure. Therefore, future validation should include larger and more balanced datasets, particularly normal-sinus-rhythm recordings, in order to assess the system's behavior under physiologically more homogeneous conditions. Even so, the software contributes a clear framework for reproducibility in biomedical signal analysis. By defining a consistent sequence of input handling, preprocessing, probabilistic transformation, and output generation, the platform allows the same analysis to be repeated under equivalent computational conditions across heterogeneous datasets. In this way, the contribution moves beyond a standalone implementation and toward a modular, extensible, and interpretable software pipeline for Holter-ECG analysis.

 doi:10.20944/preprints202604.1069.v1

5. Conclusions

The software developed in this study demonstrates that long-term Holter-ECG signal recordings can be transformed into structured, interpretable, and reproducible analytical outputs through an integrated pipeline that combines raw-signal preprocessing, probabilistic analysis, and web-based visualization. By operating within a modular client–server architecture, the system was able to process open-access PhysioNet signal records under a unified computational workflow and generate differentiated probabilistic patterns across heterogeneous datasets. SHDB-AF was predominantly associated with acute structural patterns, the Long-Term ST Database with chronic patterns, and the MIT-BIH Normal Sinus Rhythm Database with a more distributed class behavior. Although these outputs should be interpreted as dataset-level structural agreement rather than beat-level diagnosis, they support the usefulness of the platform as a reproducible software environment for probabilistic analysis of cardiac dynamics from long-term ECG signals.

Future work will focus on extending both the analytical and software capabilities of the platform. Additional temporal segmentation strategies will be explored, including alternative starting times and different daytime and nighttime windows, in order to evaluate the stability of the outputs under varying signal observation conditions. The current probabilistic approach will also be compared with a broader range of machine learning methods, including logistic regression, random forest, support vector machines, gradient boosting, and deep learning models trained on ECG-derived structured signal features. In parallel, future development will target improved interoperability, broader dataset integration, and the evolution of the platform into a more robust software environment for research and decision-support in cardiac signal dynamics analysis.

6. Patents

Author Contributions: Conceptualization, LJRL and JRV.; methodology, JRV.; software, DSAR and LJRL.; validation, LJRL; formal analysis, JRV and DSAR; investigation, LJRL.; resources, LJRL; data curation, DSAR; writing—original draft preparation, DSAR and LJRL.; writing—review and editing, DSAR and LJRL; visualization, DSAR; supervision, LJRL and DSAR; project administration, LJRL; funding acquisition, LJRL.

Funding: This research was funded by Universidad Militar Nueva Granada, grant number IMP-ING-3913 and The APC was funded by LJRL.

Acknowledgments: First, we express our sincere gratitude to the Universidad Militar Nueva Granada for the financial support provided for the development of the research project IMP-ING-3913 and Resolution 1423 of 21-07-2025.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

AF — Atrial Fibrillation
API — Application Programming Interface
BPM — Beats Per Minute
CDSS — Clinical Decision Support Systems
CSV — Comma-Separated Values
ECG — Electrocardiogram
GUI — Graphical User Interface
HRV — Heart Rate Variability
HTTP — Hypertext Transfer Protocol
JSON — JavaScript Object Notation
MIT-BIH NSRDB — MIT-BIH Normal Sinus Rhythm Database

 doi:10.20944/preprints202604.1069.v1

PHP — Hypertext Preprocessor
POST — Hypertext Transfer Protocol POST Method
RRI — R-R Interval
SHDB-AF — Japanese Holter ECG Database of Atrial Fibrillation
ST — ST Segment

References

1. World Health Organization. Cardiovascular diseases (CVDs). Available online: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds) (accessed on 11 April 2026).
2. Horsky, J.; Schiff, G.D.; Johnston, D.; Mercincavage, L.; Bell, D.; Middleton, B. Interface design principles for usable decision support: A targeted review of best practices for clinical prescribing interventions. J. Biomed. Inform. 2012, 45, 1202–1216.
3. Bayor, A.A.; Li, J.; Yang, I.A.; Varnfield, M. Designing Clinical Decision Support Systems (CDSS)—A User-Centered Lens of the Design Characteristics, Challenges, and Implications: Systematic Review. J. Med. Internet Res. 2025, 27, e63733.
4. Brito, J.J.; Li, J.; Moore, J.H.; Greene, C.S.; Nogoy, N.A.; Garmire, L.X.; Mangul, S. Recommendations to enhance rigor and reproducibility in biomedical research. GigaScience 2020, 9, giaa056.
5. Sattar, Y.; Chhabra, L. Holter Monitor. In StatPearls; StatPearls Publishing: Treasure Island, FL, USA, 2025.
6. Giada, F.; Bartoletti, A. Value of ambulatory electrocardiographic monitoring in syncope. Cardiol. Clin. 2015, 33, 361–366.
7. Diemberger, I.; Martignani, C.; Biffi, M.; Boriani, G. Holter ECG for pacemaker/defibrillator carriers: What is its role in the era of remote monitoring? Heart 2015, 101, 1272–1278.
8. Goldberger, A.L.; West, B.J. Fractals in physiology and medicine. Yale J. Biol. Med. 1987, 60, 421–435.
9. Goldberger, A.L.; Amaral, L.A.N.; Hausdorff, J.M.; Ivanov, P.C.; Peng, C.-K.; Stanley, H.E. Fractal dynamics in physiology: Alterations with disease and aging. Proc. Natl. Acad. Sci. USA 2002, 99, 2466–2472.
10. Huikuri, H.V.; Mäkikallio, T.H.; Peng, C.-K.; Goldberger, A.L.; Hintze, U.; Møller, M. Fractal correlation properties of R-R interval dynamics and mortality in patients with depressed left ventricular function after an acute myocardial infarction. Circulation 2000, 101, 47–53.
11. Rodríguez, J.O. Entropía proporcional de los sistemas dinámicos cardiacos: Predicciones físicas y matemáticas de la dinámica cardiaca de aplicación clínica. Rev. Colomb. Cardiol. 2010, 17, 115–129.
12. Rodríguez, J.; Correa, C.; Ortiz, L.; Prieto, S.; Bernal, P.; Ayala, J. Evaluación matemática de la dinámica cardiaca con la teoría de la probabilidad. Rev. Mex. Cardiol. 2009, 20, 183–189.
13. Fatovich, D.M.; Phillips, M. The probability of probability and research truths. Emerg. Med. Australas. 2017, 29, 242–244.
14. Upshur, R.E.G. A short note on probability in clinical medicine. J. Eval. Clin. Pract. 2013, 19, 463–466.
15. Velásquez, J.R.; Ramírez López, L.J.; Torres, S.G. New Predictive Diagnostic Method for Cardiac Dynamics Based on Probability Distributions. Diagnostics 2025, 15, 650.
16. Goldberger, A.L.; Amaral, L.A.N.; Glass, L.; Hausdorff, J.M.; Ivanov, P.C.; Mark, R.G.; Mietus, J.E.; Moody, G.B.; Peng, C.-K.; Stanley, H.E. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation 2000, 101, e215–e220.
17. Tsutsui, K.; Biton Brimer, S.; Ben-Moshe, N.; Sellal, J.M.; Oster, J.; Mori, H.; Ikeda, Y.; Arai, T.; Nakano, S.; Kato, R.; et al. SHDB-AF: A Japanese Holter ECG database of atrial fibrillation. Sci. Data 2025, 12, 454.
18. Jager, F.; Taddei, A.; Moody, G.B.; Emdin, M.; Antolic, G.; Dorn, R.; Smrdel, A.; Marchesi, C.; Mark, R.G. Long-term ST database: A reference for the development and evaluation of automated ischaemia detectors and for the study of the dynamics of myocardial ischaemia. Med. Biol. Eng. Comput. 2003, 41, 172–183.
19. Moody, G.B. MIT-BIH Normal Sinus Rhythm Database v1.0.0. *PhysioNet* 1999. Available online: https://www.physionet.org/physiobank/database/nsrdb/ (accessed on 11 April 2026).
20. Rodríguez Velásquez, J.R.; Oliveros Acosta, D.; Rodríguez Peña, D.S.; Sosa Pinzón, J.A.; Prieto Bohórquez, S.E.; Correa Herrera, S.C. Diagnostic methodology of cardiac dynamics based on the Zipf–Mandelbrot law: Evaluation with 50 patients. Rev. Mex. Cardiol. 2018, 29, 83–89.

 doi:10.20944/preprints202604.1069.v1

21. Rodríguez, J.; Oliveros, D.; Correa, C.; Prieto, S. Aplicabilidad clínica de software diagnóstico de la dinámica cardíaca basado en la Ley de Zipf-Mandelbrot. Rev. Haban. Cienc. Méd. 2019, 18, 624–633.
22. Laplace, P.S. Théorie analytique des probabilités; Courcier: Paris, France, 1812.
23. Kolmogorov, A.N. Grundbegriffe der Wahrscheinlichkeitsrechnung; Springer: Berlin, Germany, 1933.
24. Bernoulli, J. Ars Conjectandi; Thurnisius: Basel, Switzerland, 1713.
25. Mason, H.T.; Martinez-Cedillo, A.P.; Vuong, Q.C.; Garcia-de-Soria, M.C.; Smith, S.; Geangu, E.; Knight, M.I. A Complete Pipeline for Heart Rate Extraction from Infant ECGs. Signals 2024, 5, 118–146.
26. Badr, A.; Badawi, A.; Rashwan, A.; Elgazzar, K. XBeats: A Real-Time Electrocardiogram Monitoring and Analysis System. Signals 2022, 3, 189–208.