Energy Efficient Arrhythmia Classifier Based on Event-Driven Data

Yixiao Zhao

A Thesis Submitted to the Faculty of Graduate Studies

In Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science

Graduate Program in Electrical Engineering and Computer Science

York University

Toronto, Ontario

April 2026

Abstract

Electrocardiogram (ECG) monitoring is important for early detection of cardiac arrhythmias, but traditional uniformly sampled ECG acquisition often leads to high power consumption in long-term wearable systems. Level-crossing (LC) sampling provides an event-driven representation that can reduce redundant data. This thesis investigates heartbeat classification using LC ECG signals.

First, a feature-based classification framework is developed. ECG fiducial points are detected from LC signals, and morphological, rhythm, and LC dynamic features are constructed. After feature optimization, a multilayer perceptron classifier achieves an accuracy of 99.4% with a Macro-F1 of 0.949 on the MIT-BIH database.

Second, an event-driven spiking neural network (SNN) classifier is proposed. A time-aware leaky integrate-and-fire neuron model is introduced to incorporate event intervals into membrane dynamics. The proposed SNN achieves 99.02% accuracy and 0.936 Macro-F1, while improving the recognition of fusion beats. The results demonstrate the potential of event-driven ECG classification for low-power wearable monitoring systems.

Acknowledgements

As this thesis comes to completion, I would like to take this opportunity to express my sincere gratitude to all those who have supported, guided, and encouraged me throughout this journey.

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Yong Peter Lian, for his patient guidance, continuous support, and invaluable advice throughout my research and writing process. From the development of research ideas to the refinement of this thesis, his insight, encouragement, and rigorous academic attitude have greatly influenced me and helped me grow as a researcher.

I would also like to sincerely thank Professor Antoine Frappé, whom I had the privilege to work with during my exchange in France. His guidance and support during that period provided me with a valuable opportunity to experience a different academic environment and broaden my research perspective. That experience has been both meaningful and inspiring to me.

I am also deeply grateful to the members of my defense committee, Professor Hui Jiang and Professor Hossein Kassiri, for taking the time to review my thesis and participate in my defense. Their valuable comments and suggestions have been very helpful in improving this work.

My sincere thanks also go to my colleague Mathieu Chêne during my exchange period. His support, helpful discussions, and kindness made both my research work and daily life in France much smoother and more enjoyable. I truly appreciate the opportunity to have learned and worked alongside him.

I would also like to thank my lab mate Nhat Tri Vo for his help and support throughout this work. His assistance made the progress of my research much smoother.

Finally, I would like to express my heartfelt gratitude to my family. Their unconditional love, understanding, encouragement, and support have always been my greatest source of strength. During moments of stress and difficulty, it was their trust and companionship that gave me the confidence to keep moving forward. This thesis would not have been possible without their constant support.

To everyone who has helped me in one way or another during my studies and research, I offer my sincere thanks.

Contents

List of Tables

List of Figures

List of Acronyms

CVD	Cardiovascular Disease
ECG	Electrocardiogram
LC	Level Crossing
AAMI	Association for the Advancement of Medical Instrumentation
N	Non-ectopic Beats
S	Supraventricular Ectopic Beats
V	Ventricular Ectopic Beats
F	Fusion Beats
Q	Unclassified Beats
MIT-BIH	MIT-BIH Arrhythmia Database
PTB	Physikalisch-Technische Bundesanstalt
CPSC	China Physiological Signal Challenge
TP	True Positives
TN	True Negatives
FP	False Positive
FN	False Negatives
1D-CNN	One-dimensional Convolutional Neural Network
RNN	Recurrent Neural Network
MLP	Multilayer Perceptron
SNN	Spiking Neural Network

LIF	Leaky Integrate-and-Fire
BPTT	Backpropagation Through Time
LSTM	Long Short-Term Memory
SHAP	SHapley Additive exPlanations
PC	Personal Computer
GPU	Graphics Processing Unit
CR	Compression Ratio
SDR	Signal-to-Distortion Ratio
CDGS	Conditional Data Grouping Scheme
ANN	Artificial Neural Network
DNN	Deep Neural Network
LC-ADC	Level-Crossing Analog-to-Digital Converter

1 Introduction

Cardiovascular diseases (CVDs) remain the leading cause of death worldwide and continue to place a significant burden on healthcare systems. Early detection of cardiac abnormalities is essential for improving patient outcomes and reducing long-term medical costs. In recent years, in-home health monitoring has gained increasing attention as a practical and cost-effective approach, allowing continuous observation of patients outside traditional clinical settings.

Wearable electrocardiogram (ECG) devices have become a widely adopted solution for continuous heart monitoring. These devices can capture cardiac electrical activity in real time and provide valuable information for detecting arrhythmias and other cardiovascular conditions. Advances in wearable technology have also improved device comfort, size, and usability, making long-term daily monitoring more feasible.

Despite these advantages, continuous wearable ECG monitoring still faces important challenges. Wearable devices are typically constrained by limited battery capacity and computational resources. In many systems, ECG data are continuously transmitted to external devices or cloud servers for analysis, which leads to significant energy consumption due to wireless communication.

To address this issue, it is desirable to reduce the amount of transmitted data by performing local signal processing on the device. In this context, heartbeat classification plays a critical role. By identifying normal and abnormal heartbeats locally, the system can avoid unnecessary data transmission and significantly improve energy efficiency.

This thesis focuses on developing energy-efficient arrhythmia classification methods for wearable ECG systems. In particular, it investigates how event-driven signal representation and lightweight models can be used to reduce data redundancy and computational complexity while maintaining reliable classification performance.

1.1 Background

Cardiovascular diseases (CVDs) refer to a group of disorders affecting the heart and blood vessels, including coronary artery disease, stroke, heart failure, arrhythmia, and other conditions related to circulatory system dysfunction. As one of the most significant categories of chronic diseases, CVDs are characterized by high prevalence, long-term management, frequent clinical visits, and a high risk of disability. Therefore, they have long been a central concern in global public health. According to the World Health Organization, CVDs remain the leading cause of death worldwide, accounting for approximately 19.8 million deaths each year, which represents about 32% of all global deaths [1]. Figure 1 shows the global share of deaths by cause in 2023 [2]. This statistic highlights the severity of CVDs and indicates that they are among the most critical health challenges faced globally.

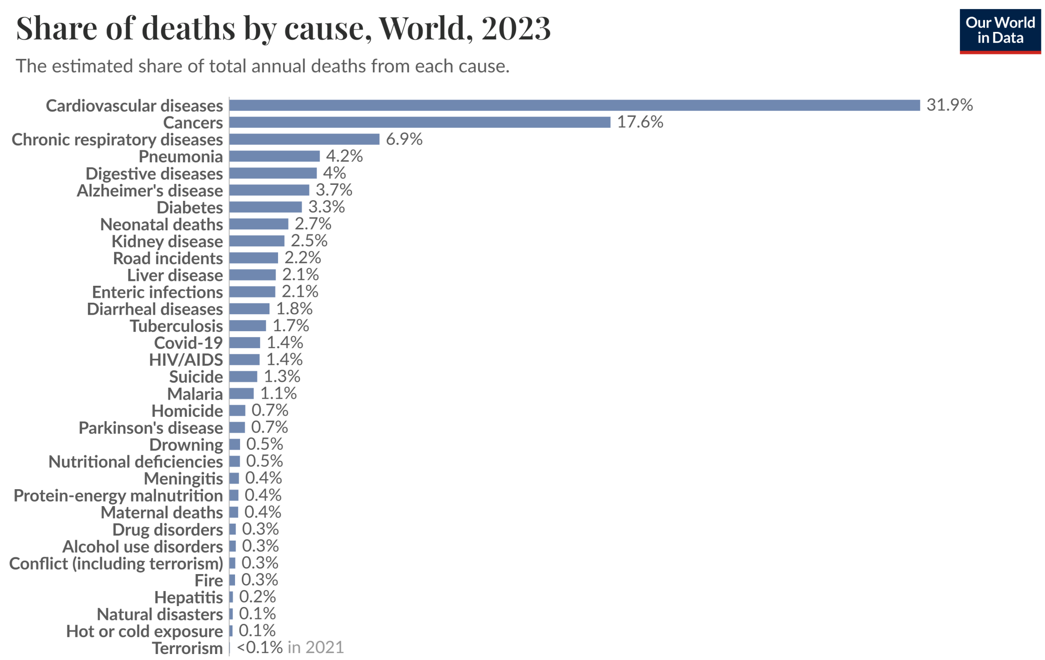

Figure 1: Global share of deaths by cause in 2023.

From the perspective of disease burden, the impact of CVDs extends beyond mortality alone. With the ongoing aging of the population, the increasing prevalence of metabolic risk factors such as hypertension, diabetes, and hyperlipidemia, as well as changes in lifestyle, the number of individuals affected by CVDs continues to rise. For healthcare systems, CVDs often require long-term medication, repeated examinations, hospitalization, and rehabilitation, resulting in substantial resource consumption. In cases where early-stage abnormalities are not detected in time, the condition may progressively worsen and eventually require emergency care or hospitalization, further increasing the burden on healthcare resources.

Taking Canada as an example, statistical reports indicate that heart disease is the second leading cause of death in the country, causing over 55,000 deaths annually. In addition, approximately 8.2% of adults are diagnosed with ischemic heart disease [3]. Considering both prevalence and mortality, it can be observed that even in developed countries with relatively well-established healthcare systems, CVDs still impose a long-term burden on healthcare infrastructure. Furthermore, CVDs are associated with significant economic costs.

According to reports from the Heart and Stroke Foundation, cardiovascular diseases result in approximately 21.2 billion Canadian dollar in direct and indirect costs annually [4]. These costs include not only medical expenditures, but also productivity loss and economic impact due to reduced workforce participation and premature death.

Therefore, reducing the burden of CVDs relies heavily on early detection, timely intervention, and long-term management. However, many cardiovascular abnormalities present as intermittent or asymptomatic changes in the early stages, making them difficult to detect using conventional short-term diagnostic methods. By the time noticeable symptoms such as chest pain, shortness of breath, fatigue, or syncope appear, the condition may have already progressed to a more severe stage, leading to higher treatment costs and reduced effectiveness of intervention. In contrast, continuous monitoring of cardiac activity in daily life can provide opportunities to detect abnormalities at an earlier stage, thereby supporting more timely diagnosis and intervention.

Among various cardiac monitoring methods, the electrocardiogram (ECG) is one of the most mature and widely used techniques. An ECG records the electrical activity of the heart over time and reflects the electrophysiological processes within each cardiac cycle. A typical ECG waveform consists of several key components, including the P wave, QRS complex, and T wave. Figure 2 shows the classical ECG curves with its most common waveforms [5]. The P wave corresponds to atrial depolarization, the QRS complex represents ventricular depolarization, and the T wave reflects ventricular repolarization. In addition to waveform morphology, temporal intervals such as the RR interval, QRS duration, and QT interval provide important information about heart rhythm, conduction, and cardiac function. Therefore, ECG signals serve as a fundamental basis for arrhythmia detection and cardiovascular assessment.

From the perspective of signal acquisition, clinical ECG systems commonly employ multi-lead configurations, such as the standard 12-lead ECG. Figure 3 shows 12-lead ECG lead placement [6]. These systems record cardiac electrical activity from multiple spatial directions by placing electrodes at different locations on the body. Each lead can be interpreted as observing the heart's electrical activity from a specific angle, resulting in variations in waveform shape and amplitude across different leads. The combination of limb leads and chest leads provides a comprehensive representation of cardiac activity, enabling clinicians to detect localized abnormalities such as myocardial ischemia or conduction disorders that may only appear prominently in certain leads.

However, multi-lead ECG systems typically require multiple electrodes, complex setup procedures, and

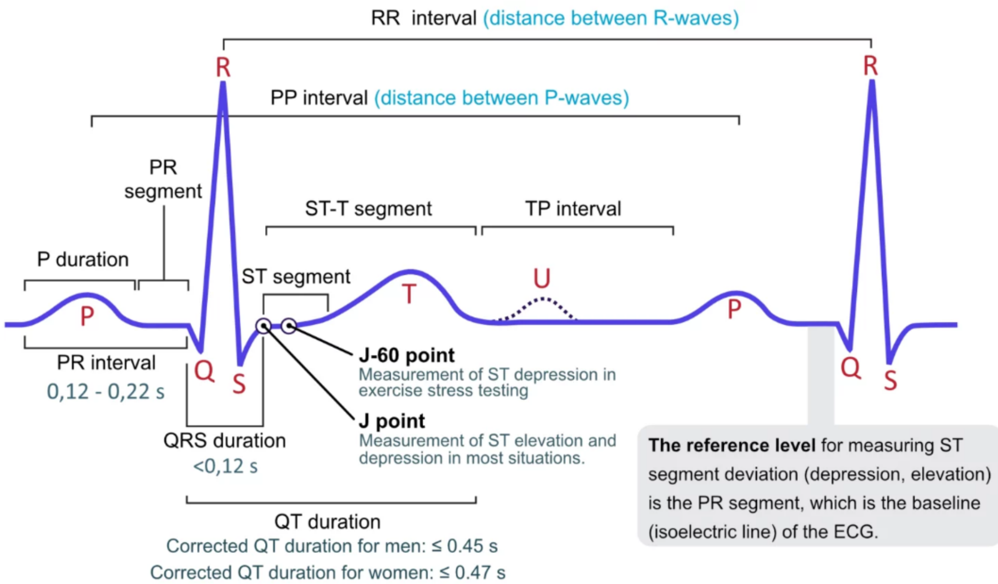

Figure 2: The classical ECG curves with its most common waveforms.

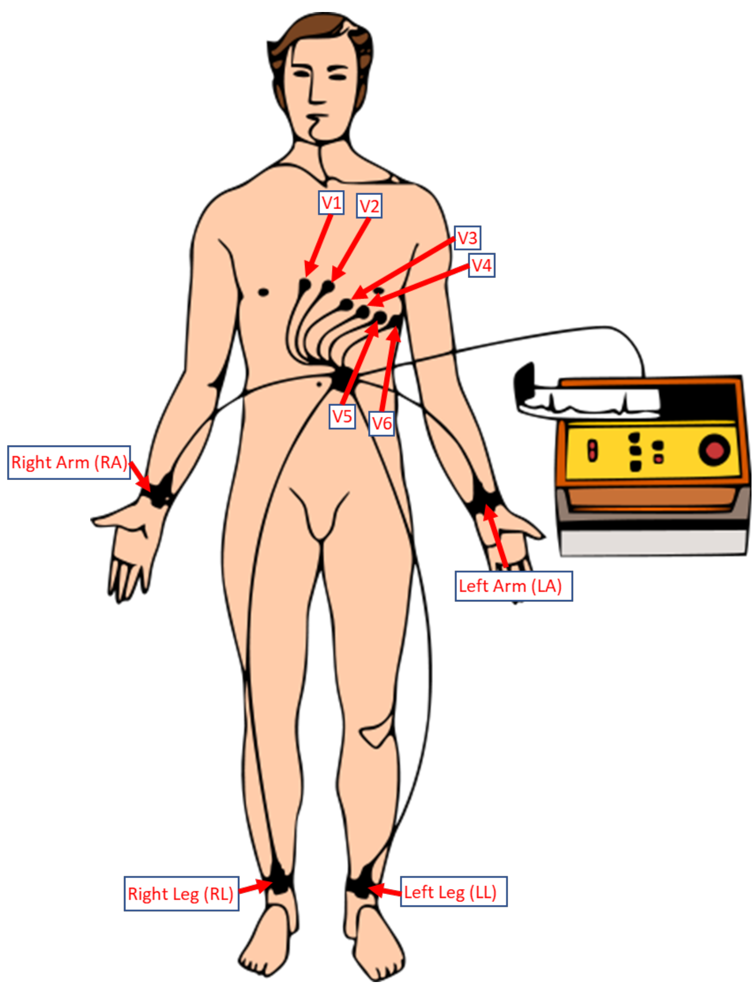

Figure 3: Annotated 12-Lead ECG lead placement.

professional operation, making them more suitable for short-term clinical examinations rather than long-term continuous monitoring. In contrast, single-lead or few-lead ECG systems are more practical for wearable and home monitoring applications. Although they provide less spatial information compared to 12-lead systems, they are often sufficient for rhythm analysis, R-peak detection, heartbeat segmentation, and basic arrhythmia screening. Moreover, such systems offer advantages in terms of reduced hardware complexity, lower power consumption, and improved user comfort, making them well suited for long-term continuous monitoring.

Traditional long-term ECG monitoring is commonly performed using Holter monitoring systems, which typically record ECG signals continuously for 24 to 48 hours. A typical Holter monitor used for ECG recording is shown in Figure 4 [7]. During monitoring, patients wear multiple electrode patches connected via wires to a portable recording device that continuously stores ECG data. Compared to short-duration clinical ECG recordings, Holter monitoring increases the likelihood of capturing transient or intermittent arrhythmias over an extended observation period.

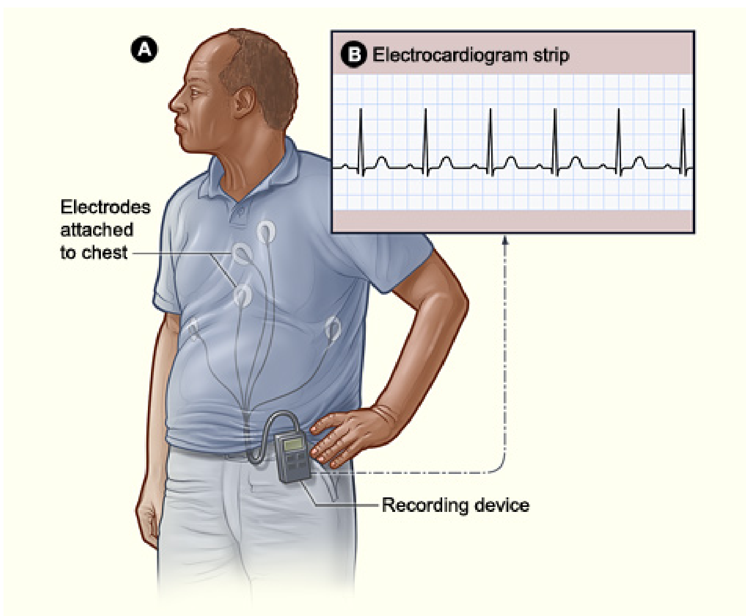

Figure 4: Holter monitor used for ECG recording.

Despite its advantages, Holter monitoring still has several limitations. First, the monitoring duration is generally limited to 24 to 48 hours, which may be insufficient for detecting low-frequency or sporadic arrhythmic events. Second, the requirement of wearing electrodes, wires, and recording devices may cause discomfort and inconvenience, reducing patient compliance, especially for prolonged use. In addition, Holter systems typically operate in an offline manner, where data are analyzed only after the recording period is completed, limiting their ability to provide real-time feedback or early warning.

Given these limitations, wearable ECG devices have emerged as a promising solution for cardiovascular monitoring. Compared to traditional clinical systems, wearable devices enable continuous ECG acquisition in daily life environments, allowing long-term and real-world monitoring of cardiac activity. This approach increases the probability of capturing abnormal events and supports applications such as remote healthcare, chronic disease management, and early risk detection. Furthermore, by shifting part of the monitoring process from hospitals to home environments, wearable systems offer the potential to alleviate pressure on healthcare resources while improving accessibility and convenience of cardiac monitoring.

1.2 Motivation and Problem Statement

In the context described above, bringing ECG monitoring from clinical settings into daily life changes the nature of the design problem. Traditional clinical ECG is usually recorded in a relatively controlled environment, over a limited time window, and can be analyzed after acquisition. Wearable ECG devices, by contrast, are expected to operate continuously during ordinary activities such as working, resting, and sleeping. As a result, the central issue is no longer simply whether ECG signals can be acquired, but whether meaningful monitoring can be sustained over long periods under limited system resources [8].

For wearable ECG systems, the most immediate constraint comes from the device itself. In order to support long-term use, the device must remain lightweight, compact, and comfortable to wear. These requirements directly limit battery size and, consequently, the available energy budget. At the same time, long-term monitoring requires the system to operate continuously for many hours or even days. This mismatch between limited energy supply and prolonged operation makes low-power design a fundamental requirement rather than a secondary optimization goal [8].

A typical wearable ECG monitoring system is shown in Figure 5. It includes signal acquisition, data processing, and wireless transmission. Although these three components work together, they do not contribute equally to overall energy consumption. The front-end acquisition circuit usually has relatively stable power consumption, while the computational cost depends on the complexity of the processing algorithm. Wireless communication, however, often becomes a major burden during long-term operation, because energy is consumed not only in sending data but also in maintaining the communication protocol and link activity [9]. For this reason, system lifetime is closely related to how much data must be transmitted over time.

Figure 5: A typical wearable ECG monitoring system.

The problem is further aggravated by the nature of ECG data itself. Continuous ECG monitoring produces a large amount of raw data, but the diagnostic value of these data is not uniformly distributed. Long segments of the waveform may vary slowly and contain relatively little information relevant to arrhythmia detection, whereas clinically important information is often concentrated around specific beats and their local structures. In wearable settings, one of the practical difficulties is therefore not only the total amount of data, but also how to identify the portion of the signal that is actually worth preserving, processing, or transmitting [10].

Many existing wearable ECG systems still rely on a straightforward architecture in which the sensor node mainly performs signal acquisition, while most of the analysis is delegated to a smartphone, gateway, or cloud platform. This design simplifies the sensor-side implementation, but it also creates a persistent communication demand. For short-duration monitoring, this may be acceptable. For long-term continuous monitoring, however, the repeated transmission of raw waveform data can quickly become a limiting factor for battery life [9].

Under this setting, incorporating a certain degree of local intelligence at the device side becomes a natural direction. Instead of transmitting all raw ECG signals, the system can first perform a preliminary analysis locally and then decide which information is worth sending out. In this framework, heartbeat classification becomes particularly important. Although ECG appears as a continuous time series, its diagnostic meaning is largely associated with individual beats. Many arrhythmias manifest themselves through abnormal beat morphology or rhythm patterns. If the system can classify each heartbeat at the device side, then the continuous waveform can be converted into event-level information that is more directly useful for subsequent decisions [11].

The significance of classification here is therefore not limited to assigning labels to beats. More importantly, it enables the system to change how data are handled. If the detected beats are normal, the upload frequency may be reduced and only essential information needs to be retained. If abnormal beats are detected, then further analysis, data transmission, or alarm generation can be triggered. In this sense, classification supports a more selective processing pipeline, allowing the system to focus its resources on clinically meaningful events rather than treating all data equally [11].

From a system-design perspective, this reflects a trade-off between local computation and wireless communication. As long as the local classification module remains sufficiently lightweight, the additional processing cost may be justified by the reduction in communication demand. This does not mean that the communication cost is quantitatively modeled in this thesis; rather, it motivates the design choice that efficient on-device analysis can support more resource-aware ECG monitoring in wearable settings [9].

At the same time, moving classification to the device side introduces new challenges. First, the model must be small enough to fit within the limitations of embedded hardware, including processing capability, memory capacity, and power budget. Second, classification performance must remain reliable, especially for abnormal or infrequent heartbeat categories. Third, wearable ECG signals are collected in real-world environments, where motion, electrode contact variation, and sensor displacement may introduce fluctuations in signal quality. These factors increase the practical difficulty of beat-level analysis, even when robustness itself is not the primary focus of this work [8].

Current research has already shown that on-device ECG classification is feasible, but it also makes clear that feasibility depends on the balance among accuracy, complexity, latency, and deployment cost. A recent lightweight single-lead ECG classification study reported 96.32% accuracy while compressing the student model to 1/1242.58 of the teacher model, demonstrating that substantial model reduction is possible for embedded deployment [12]. Another recent on-chip arrhythmia classifier implemented on a resource-constrained wearable platform achieved 98.09% accuracy, 20 ms inference latency, and 17.7 mW model power, further illustrating that real-time beat-by-beat classification can be realized under strict hardware constraints [13].

At the same time, recent reviews emphasize that the key concern in low-power ECG classification is no longer classification accuracy alone, but the broader question of real-time operation, low complexity, and deployability in resource-constrained environments [11]. This means that, for wearable ECG systems, classification should not be considered as an isolated algorithmic problem. Instead, it must be treated as part of a larger system design problem involving data representation, model complexity, and long-term operating conditions.

Accordingly, the core problem addressed in this thesis is not simply how to maximize classification accuracy, but how to achieve effective and reliable beat classification under limited resources through more efficient data representation and more lightweight model design. This problem involves both the reduction of redundant input information and the development of models that are suitable for embedded deployment. Based on this motivation, the objective of this thesis is to develop an arrhythmia classification approach for wearable ECG applications that balances classification performance with implementation efficiency, thereby supporting long-term and low-power cardiac monitoring.

1.3 Research Objectives

Based on the background and problem analysis presented above, the overall objective of this thesis is to develop an arrhythmia classification approach for wearable ECG applications that balances classification performance with implementation efficiency, so that it is more suitable for long-term and resource-constrained monitoring systems.

To achieve this objective, this thesis focuses on two closely related issues. First, long-term ECG monitoring generates a large amount of raw data, much of which is redundant. Therefore, a more compact data representation is needed so that the input can preserve discriminative information while reducing the burden on subsequent processing. Second, the classification model itself must provide reliable recognition performance while keeping its complexity under control, so that it is more compatible with practical deployment on wearable devices.

Accordingly, the specific objectives of this research are as follows.

First, this research investigates data representation methods for ECG heartbeat classification, with the aim of organizing the input in a way that more directly reflects information relevant to arrhythmia recognition, rather than relying entirely on full waveform input. By doing so, the input can be made more compact and efficient for subsequent classification.

Second, this research studies beat-level classification methods that can distinguish between normal and abnormal heartbeats in continuous ECG signals. In this context, the focus is not limited to overall classification accuracy, but also includes the model's ability to recognize different abnormal heartbeat categories, since performance on abnormal classes is particularly important in arrhythmia detection.

Third, this research investigates classification models with controlled complexity. The goal is to maintain effective classification performance while reducing model size and computational cost, so that the resulting method is more suitable for resource-constrained environments. Rather than pursuing accuracy alone, this work emphasizes a more practical balance between recognition capability and implementation cost.

Fourth, this research evaluates the proposed methods on public ECG datasets and analyzes both their classification performance and implementation-related cost. The purpose is to demonstrate that meaningful arrhythmia classification can still be achieved without relying on excessively large or computationally intensive models.

Under these objectives, the research is developed along two technical directions. The first direction is a feature-based classification framework, which aims to achieve effective beat classification through compact input representation and lightweight models. The second direction further explores a classification framework that is more compatible with event-driven processing. Although these two directions differ in methodology, they are motivated by the same goal: to reduce redundancy, control complexity, and support practical arrhythmia classification for wearable ECG systems.

Overall, this research seeks to answer the following question: in wearable ECG applications, how can reliable arrhythmia classification be achieved under limited resources through more suitable data representation and more efficient model design? The answer to this question is directly related not only to classification performance itself, but also to whether such methods can realistically support long-term and low-power cardiac monitoring.

1.4 Thesis Organization

The remainder of this thesis is organized as follows.

Chapter 2 reviews the basic characteristics of ECG signals, common types of arrhythmias, and the main challenges involved in long-term wearable ECG monitoring. It also summarizes existing approaches related to ECG classification, with particular attention to methods that are relevant to low-power and resource-constrained applications.

Chapter 3 presents the first technical direction of this thesis, namely a feature-based heartbeat classification framework. This chapter describes the data representation used for beat-level analysis, the design of the corresponding feature set, and the lightweight classification model built on top of it. Experimental results are then provided to evaluate the effectiveness of this approach.

Chapter 4 presents the second technical direction of this thesis, which further explores a classification framework that is more compatible with event-driven processing. This chapter focuses on how the ECG signal can be represented and processed in a different way so as to better support efficient classification under resource constraints. The proposed model and its experimental evaluation are also given in this chapter.

Finally, Chapter 5 concludes the thesis by summarizing the main findings of this work, discussing its limitations, and outlining possible directions for future research.

2 Related Work on Low-Power ECG Classification

This chapter builds upon the previously introduced research background and problem statement, and provides a systematic review of the fundamental concepts and existing studies related to automatic ECG classification. The purpose is to establish a clear theoretical background and problem scope for the subsequent method development. Specifically, this chapter first introduces the main abnormal heartbeat types involved in ECG classification and the common task formulations, explaining why heartbeat-level classification has become one of the most widely adopted settings in this field. It then reviews commonly used public datasets, label mapping strategies, and evaluation protocols, thereby clarifying the unified task foundation for subsequent method comparisons.

Building on this foundation, the chapter further summarizes the development of existing ECG classification approaches, including early methods based on handcrafted features and conventional machine learning, as well as the deep learning methods that have become dominant in recent years. Their characteristics are discussed in terms of classification performance, representation capability, and model complexity. Finally, the chapter reviews the current research status of low-power and on-chip ECG classification, with a focus on the remaining challenges in input data scale, model lightweighting, and system-level deployment feasibility. Through this analysis, this chapter highlights that, although existing methods have achieved promising performance on public datasets, their direct application in wearable and resource-constrained scenarios remains challenging, thereby motivating the subsequent studies on input representation optimization and lightweight classification design.

2.1 ECG Abnormality Types and Classification Formulation

The basic components of ECG signals and the normal cardiac cycle have already been introduced in Chapter 1, and therefore they are not repeated in detail here. For the purpose of the present study, what needs to be clarified is which types of abnormalities are considered in automatic classification tasks, how these abnormalities are typically reflected in ECG signals, and why most existing studies adopt beat-level classification rather than rhythm-level classification over a longer segment.

From the perspective of task formulation, automatic ECG analysis can generally be divided into two categories. One is *rhythm-level classification*, in which the overall rhythm state is determined within a relatively long time window. For example, a segment lasting several seconds or longer may be classified as sinus rhythm, atrial fibrillation, atrial flutter, or another sustained rhythm abnormality. The other is *beat-level classification*, in which a continuous ECG recording is segmented into individual heartbeats and each beat is classified separately. The former emphasizes global rhythm patterns, whereas the latter focuses more on the local morphology of each beat and its temporal relationship with neighboring beats.

For wearable ECG applications and automatic arrhythmia detection, beat-level classification is particularly important. Many abnormalities do not persist throughout an entire recording; instead, they appear as premature beats, waveform distortion, or altered conduction patterns in only a few individual heartbeats. If the analysis is performed only over a relatively long time window, the system requires large on-chip memory to store raw data which increases the area and power. In contrast, beat-level analysis makes it easier to capture these clinically meaningful local changes. For this reason, heartbeat classification studies usually begin with R-peak detection, followed by segmentation of individual beats centered around the detected R peaks, and the classification target is then defined at the single-beat level.

In this line of research, the original heartbeat labels provided in public databases are often numerous, fine-grained, and highly imbalanced. To facilitate comparison among different methods, many studies reorganize the original labels according to the *AAMI* standard. AAMI stands for the *Association for the Advancement of Medical Instrumentation*. In ECG classification research, the AAMI recommendation is commonly used to group the original heartbeat labels into a smaller number of categories. It should be emphasized that, in this research, the AAMI grouping is not intended to represent a complete clinical diagnosis. Rather, it serves as a standardized labeling framework for automatic classification research, so that different studies can report results under a relatively consistent category setting instead of using overly complicated original annotations.

Under this standardized setting, heartbeats are commonly organized into five classes: N, S, V, F, and Q. Figure 6 shows different types of heartbeats annotated with clinical features [14]. Although these classes originate from a data-labeling protocol, they still correspond to different forms of cardiac electrical activity, and therefore it is necessary to explain their medical characteristics.

The *N class* usually refers to *non-ectopic beats*, which may be understood as normal beats or beats generated under near-normal conduction conditions. These beats originate from the sinus node and propagate through the normal cardiac conduction system. Accordingly, the ECG usually shows a relatively stable rhythm, a clear relationship between the P wave and the QRS complex, a normal QRS duration, and an overall regular waveform morphology. In automatic classification, the N class is usually the largest category

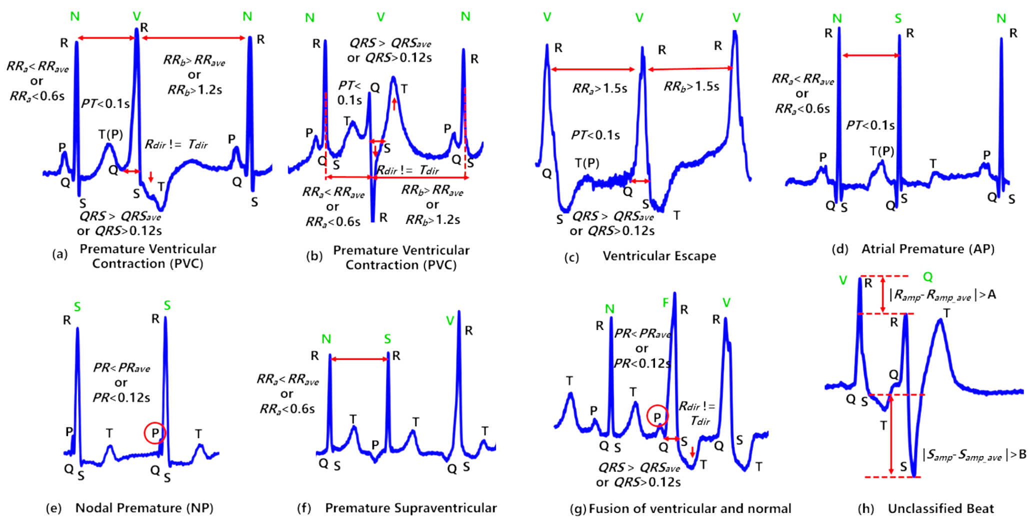

Figure 6: Different types of heartbeats annotated with clinical features.

in the dataset and also serves as the most important reference when identifying other abnormal beat types. In many cases, the decision about whether a beat is abnormal is essentially based on how it differs from this normal pattern.

The *S class* represents *supraventricular ectopic beats*. These beats originate from a location in the atria or above the atrioventricular junction, and therefore the abnormality first occurs at the level of where the excitation begins rather than in ventricular conduction itself. A typical example is the premature atrial contraction. On the ECG, such a beat often appears earlier than expected, meaning that it occurs before the time predicted by the normal rhythm. At the same time, because the excitation source is altered, the morphology of the P wave is often different from that of a normal sinus P wave. However, since the impulse can still usually travel to the ventricles through the normal atrioventricular conduction system, the QRS complex often remains narrow and does not become markedly widened as in ventricular abnormalities. For this reason, the difficulty in recognizing S-class beats does not primarily lie in a severely distorted QRS complex, but rather in the fact that the beat occurs prematurely, the P wave looks different, and the temporal relationship between neighboring beats is disturbed. From a classification perspective, supraventricular ectopic beats are often harder to distinguish than ventricular ectopic beats because their local waveform may still resemble a normal beat.

The *V class* represents *ventricular ectopic beats*, with the premature ventricular contraction being the most typical example. The key difference between the V class and the S class is that these beats do not originate in the atria or near the atrioventricular node, but directly within the ventricles. As a result, depolarization no longer propagates rapidly through the normal His–Purkinje system, but instead spreads more slowly through the ventricular myocardium. This leads to a very characteristic ECG appearance: the beat usually occurs prematurely, while the QRS complex becomes markedly widened and morphologically abnormal. In some cases, no normal P wave can be seen before the beat; even if a P wave is present, it may not maintain a normal conduction relationship with the corresponding QRS complex. In addition, the waveform of a V-class beat is often broader, blunter, and more distorted than that of a normal beat, and the T-wave direction may also change. For automatic classification, V-class beats are often easier to separate from normal beats because their morphology differs more strongly. On the other hand, the waveform variation within the V class itself can also be large, which makes robust classification nontrivial.

The *F class* represents *fusion beats*. This category is somewhat more difficult to understand because such a beat is not generated independently by a single source of excitation. Instead, it is produced when two different sources of excitation contribute simultaneously to the depolarization of the same heartbeat. In other words, an impulse conducted through the normal pathway and an ectopic impulse participate together in ventricular depolarization, so that the resulting QRS complex is neither fully normal nor fully ventricular in appearance, but lies somewhere between the two. For this reason, F-class beats often appear as an intermediate morphology between normal and abnormal beats. From the perspective of automatic classification, this class is difficult because the number of samples is usually small and the class boundary is inherently vague. Rather than following one stable template, fusion beats often exhibit mixed waveform characteristics.

The *Q class* generally includes unclassified beats, beats from special sources, or beats that cannot be clearly assigned to the other categories. Unlike the previous classes, the Q class does not correspond to one particularly stable or typical medical waveform pattern. Instead, it acts more like a residual category that collects beats that are difficult to place into N, S, V, or F under the standard grouping scheme. The importance of this class lies not in having a uniform morphology, but in representing a higher degree of uncertainty. For an automatic classifier, the Q class often reflects the model's ability to generalize when dealing with ambiguous, boundary, or uncommon samples.

From a medical point of view, the reason why these heartbeat classes can be distinguished is that they correspond to different excitation origins and conduction pathways. The N class follows normal sinus initiation and normal conduction. The main change in the S class is that the excitation begins earlier and from a different supraventricular source, while ventricular conduction remains largely normal. The V class directly alters the ventricular depolarization pathway, which explains the more obvious widened and distorted QRS complex. The F class reflects the combined effect of two excitation sources, whereas the Q class mostly represents boundary cases that cannot be clearly classified within the standardized framework. For automatic classification, however, these classes are not perfectly separated. V-class beats are usually easier to distinguish from normal beats in terms of morphology, whereas the boundary between the S and N classes is often more ambiguous, and F-class beats often lie somewhere between normal and ventricular ectopic beats. In other words, the challenge of heartbeat classification is not merely to determine whether a beat is abnormal, but also to distinguish among different abnormal classes and between abnormal and normal beats when their boundaries partially overlap.

Therefore, it is necessary to describe these abnormal beat types and their ECG characteristics in this chapter. The purpose here is not to provide a full clinical course in electrocardiology, but rather to establish a clear medical background for the subsequent discussion of dataset organization, label setting, feature construction, and classification model design. Regardless of whether handcrafted features or learned representations are used later, the model ultimately attempts to distinguish among these differences in beat morphology and rhythm relationship.

2.2 Public Datasets, Label Mapping, and Evaluation Protocols

In ECG heartbeat classification research, public datasets not only provide the data foundation for model training and testing, but also directly affect whether different methods can be fairly compared. This point is particularly important for the heartbeat classification task considered in this thesis. ECG signals exhibit considerable inter-subject variability, and different databases may differ substantially in sampling rate, number of leads, recording duration, annotation granularity, and class distribution. Therefore, when reviewing previous studies, it is not sufficient to examine only the classifier architecture or the final reported performance; the adopted dataset, label organization, and evaluation protocol must also be taken into account.

Among existing datasets, the *MIT-BIH Arrhythmia Database* remains one of the most widely used benchmark datasets for beat-level ECG classification. It consists of 48 ambulatory ECG recordings with expert annotations, each approximately 30 minutes in duration and sampled at 360 Hz. Since each record provides detailed beat-level annotations, labeled heartbeat samples can be extracted relatively conveniently from the continuous signals. This is one of the main reasons why MIT-BIH has long been used as a standard benchmark for arrhythmia heartbeat classification.

The widespread use of MIT-BIH is not simply due to its historical importance, but also to several practical advantages for methodological studies. First, it provides complete and detailed heartbeat annotations, which make beat-level classification tasks relatively easy to formulate. Second, it contains a large number of normal beats as well as multiple types of abnormal beats, thereby reflecting several core challenges of ECG classification, including class imbalance, overlapping class boundaries, and limited abnormal samples. Third, because MIT-BIH has been extensively used in the literature, new methods evaluated on this dataset can be more directly compared with previous work. For a study such as this thesis, which focuses on low-power and resource-constrained scenarios, this aspect is particularly valuable, because discussions on input representation, model complexity, and implementation cost become more meaningful when they are conducted on a common benchmark.

MIT-BIH is, of course, not the only publicly available ECG dataset. In recent years, datasets such as PTB, PTB-XL, CPSC, and several long-term wearable ECG datasets have also been increasingly used in ECG-related research. These datasets differ in sample size, label definition, number of leads, and task formulation. Some are more suitable for multi-lead diagnostic classification, while others are more appropriate for long-term rhythm recognition or large-scale deep learning training. However, from the perspective of single-beat arrhythmia classification and methodological comparison, MIT-BIH remains one of the most representative benchmark datasets. The focus of this thesis is not to build a universal diagnostic system for all ECG scenarios, but rather to investigate the relationship between data representation and model complexity under a standardized heartbeat classification setting. For this purpose, the representativeness and comparability of MIT-BIH are still highly valuable.

It should also be noted that the original labels in public datasets are often not directly suitable for standardized comparison. In the case of MIT-BIH, the original annotation set contains many beat types, some of which have very few samples, and the clinical distinctions between some labels are relatively fine-grained. If all original labels were used as final classification targets, the task would become substantially more complex, and results from different studies would be difficult to compare in a consistent way. For this reason, many ECG heartbeat classification studies do not use all raw labels directly. Instead, they reorganize the original annotations into a smaller number of target categories based on widely adopted standardized settings. For heartbeat classification, the most common approach is to use the AAMI five-class setting.

The previous subsection has already introduced the main abnormal beat types represented in the AAMI five-class formulation, and therefore its background is not repeated here. What matters in the present subsection is how the original MIT-BIH annotations are mapped to the five AAMI classes. This mapping is important because it defines the actual classification problem being solved and directly affects whether the results reported in different studies are truly comparable. Two methods may use the same database, but if one directly uses the original annotations while the other first maps them into the AAMI classes, then they are not solving exactly the same task.

Table 1 presents the typical correspondence between MIT-BIH raw annotations and the AAMI five-class setting. It should be noted that minor differences may exist across individual studies, but the overall mapping strategy is largely consistent. The common practice is to merge raw labels with similar electro-physiological characteristics and similar research significance into broader target categories.

Table 1: Typical mapping between MIT-BIH raw annotations and the AAMI five-class setting.

AAMI class	**Class meaning**	**Common raw labels**
N	Non-ectopic beats	N, L, R, e, j
S	Supraventricular ectopic beats	A, a, J, S
V	Ventricular ectopic beats	V, E
F	Fusion beats	F
Q	Unknown / unclassifiable beats	/, f, Q

As can be seen from Table 1, the AAMI five-class setting is not simply a reduction in the number of labels, but a standardized reorganization of the original annotations. Its main advantages are twofold. First, it reduces task complexity, allowing researchers to focus on the discrimination among major heartbeat classes rather than dealing with extremely fine-grained raw labels with very limited sample sizes. Second, it allows different studies to report results under a relatively unified label framework. This is important for the present work because the low-power and on-chip classification methods discussed later are not intended to solve an arbitrarily complex diagnostic labeling problem, but rather to examine the effects of input representation and model complexity under a standardized and comparable heartbeat classification setting.

In addition to label mapping, the *training and testing protocol* is another essential aspect that must be clearly specified in ECG classification studies. In the existing literature, the most common settings can be broadly described as *patient-specific* and *patient-independent*; the latter is often implemented in the form of an *inter-patient* setting. Although both are used for heartbeat classification, they correspond to different research questions, and performance results can be misleading if these settings are not clearly distinguished.

In a *patient-specific* setting, part of the data from the target subject is used for training, and the remaining part of the same subject's data is used for testing. Under this setting, the model can learn subject-specific waveform patterns, rhythm characteristics, and individual variations, and therefore typically achieves higher classification performance. For personalized ECG classification, this is a reasonable setup because the objective is precisely to adapt the model to a specific user. However, its limitation is equally clear: it mainly reflects the model's ability to adapt to a *known* subject, rather than its ability to generalize to *unseen* subjects.

In contrast, a *patient-independent* or *inter-patient* setting requires the training and testing sets to come from different subjects. In other words, the model is evaluated on subjects that were never seen during training. Compared with a patient-specific setting, this is a more stringent evaluation protocol, because ECG waveforms vary significantly across individuals, and the model must rely on more general heartbeat characteristics rather than person-specific patterns. For methods intended for practical wearable applications, a patient-independent setting is usually more meaningful, because real-world systems are often expected to operate on users whose data were not available during training.

For this reason, even when the same MIT-BIH database and the same AAMI five-class setting are used, the reported performance of different studies may still differ substantially, and one important reason is the training/testing protocol. Some patient-specific methods may achieve very high accuracy, but this does not necessarily imply equally strong performance under cross-subject conditions. Conversely, some methods evaluated under an inter-patient setting may report lower accuracy, but their performance may better reflect the challenges encountered in realistic applications. Therefore, when reviewing previous work, it is not enough to consider only the final numbers; the evaluation protocol must also be examined carefully.

Another important issue is the choice of *evaluation metrics*. Because heartbeat classification often suffers from strong class imbalance, overall accuracy alone is usually insufficient to reflect actual model behavior. For instance, if the number of normal beats is much larger than that of abnormal beats, a model biased toward the majority class may still achieve a high overall accuracy while performing poorly on clinically important abnormal classes. For this reason, recent studies usually report not only accuracy, but also sensitivity, specificity, precision, and F1-score in order to describe classification performance more comprehensively.

Accuracy represents the proportion of correctly classified samples among all samples, and is defined as

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}, \tag{1}$$

where TP denotes true positives, TN denotes true negatives, FP denotes false positives, and FN denotes false negatives.

Sensitivity, also referred to as *recall*, represents the proportion of true positive samples that are correctly identified, and is defined as

$$\text{Sensitivity} = \frac{TP}{TP+FN}. \tag{2}$$

A higher sensitivity indicates fewer missed detections of the positive class. In ECG arrhythmia classification, this metric is especially important because missing abnormal beats is often more critical than producing some false alarms.

Specificity represents the proportion of true negative samples that are correctly classified as negative, and is defined as

$$\text{Specificity} = \frac{TN}{TN+FP}. \tag{3}$$

This metric reflects the model's ability to reject non-target samples. In ECG classification, poor specificity means that a large number of normal beats may be incorrectly identified as abnormal, which can lead to unnecessary downstream processing.

Precision represents the proportion of predicted positive samples that are truly positive, and is defined as

$$\text{Precision} = \frac{TP}{TP+FP}. \tag{4}$$

A low precision indicates that the model produces many false alarms among the beats predicted as positive.

The *F1-score* is the harmonic mean of precision and sensitivity, and is commonly used to provide a balanced measure between detection completeness and prediction correctness. It is defined as

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}}. \tag{5}$$

For heartbeat classification tasks with strong class imbalance, the F1-score often provides a more informative measure than accuracy alone, especially for abnormal classes.

In multi-class classification problems, these metrics can be computed for each class individually and further summarized through macro-averaging or weighted averaging. For the experiments presented later in this thesis, it is important not only to examine the overall classification results, but also to pay close attention to the performance on abnormal classes. The practical value of arrhythmia detection does not lie simply in correctly classifying a large number of normal beats, but in maintaining reliable detection capability for the relatively rare yet clinically significant abnormal beats.

Overall, public datasets, label mapping, evaluation protocols, and performance metrics together form the experimental basis of ECG heartbeat classification research. In the context of this thesis, clarifying these issues is not merely a matter of repeating standard practice. Rather, it is necessary for establishing a unified foundation for subsequent method design and result analysis. Only when the task definition, label organization, training protocol, and evaluation criteria are all clearly specified can the later discussion on input representation, model complexity, and low-power suitability remain interpretable and comparable.

2.3 Existing ECG Classification Methods

The development of ECG arrhythmia classification has generally evolved from traditional handcrafted-feature-based methods to deep learning-based methods. These two stages differ substantially in terms of input representation, feature extraction strategy, model structure, and implementation cost, and therefore correspond to different research priorities. For this thesis, reviewing these methods is not merely intended to summarize how high the reported classification performance has become on public datasets. More importantly, it is necessary to clarify how general ECG classification methods have been developed, what problems they have addressed, and what limitations remain. Recent review studies consistently show that the current literature can be broadly organized into two main streams: one based on handcrafted features combined with conventional machine learning classifiers, and the other based on automatic representation learning directly from ECG waveforms [15, 16].

Earlier ECG classification studies mainly relied on handcrafted feature extraction together with conventional machine learning models. Figure 7 shows General pipeline of ECG heartbeat classification. In such approaches, ECG signals are typically preprocessed first, including denoising, baseline wander correction, and R-peak detection. After heartbeat segmentation, discriminative morphological, interval-related, and rhythm-related features are extracted and then fed into classifiers such as support vector machines, k-nearest neighbors, decision trees, random forests, or shallow neural networks. The central idea is to transform the raw waveform into a lower-dimensional structured representation and then perform classification using a relatively simple classifier. A recent example is the work by Jahangir *et al.*, who combined feature engineering with a hybrid stacked machine learning framework for ECG arrhythmia classification and systematically compared different feature selection strategies and ensemble configurations [17].

The main advantages of traditional methods lie in their interpretability and methodological transparency. Since the extracted features often correspond directly to clinically meaningful ECG characteristics, such as RR interval variation, QRS width, and local waveform amplitudes, the resulting decisions are easier to in-

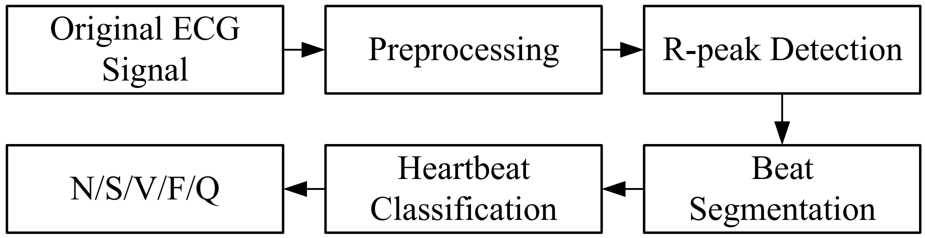

Figure 7: General pipeline of ECG heartbeat classification.

terpret from a medical perspective. These methods also tend to be relatively stable when the dataset size is limited and the task definition is well constrained. However, their weaknesses are equally clear. On the one hand, their performance depends heavily on feature engineering, and the choice of features may vary substantially from study to study. On the other hand, they are highly sensitive to upstream processing steps, such as R-peak localization and heartbeat segmentation, and inaccuracies in these stages may degrade feature quality and ultimately reduce classification performance. Therefore, while handcrafted-feature methods remain useful, their performance ceiling is often constrained by the quality and completeness of the manually designed feature set.

With the rapid progress of deep learning, an increasing number of studies have shifted toward end-to-end ECG classification models. Unlike traditional methods, deep learning approaches do not require all relevant features to be explicitly designed in advance. Instead, they learn multi-level representations directly from ECG waveforms. The most common structures include one-dimensional convolutional neural networks (1D-CNNs), recurrent neural networks (RNNs) and their variants, CNN-RNN hybrid models, and more recently, Transformer-based models. Recent reviews indicate that CNNs, LSTMs, and hybrid CNN-LSTM architectures have become the dominant frameworks in ECG arrhythmia classification, and that the main research trend has moved from simply demonstrating automatic classification to improving performance under increasingly complex task settings [15, 16].

Among existing deep learning approaches, convolutional models remain the most widely used. This is mainly because ECG signals exhibit strong local structures, such as QRS morphology, peak locations, and amplitude variations, which can be effectively captured by one-dimensional convolutions [15]. Convolutional architectures are therefore naturally suitable for heartbeat classification, where local beat morphology plays a major role in class discrimination. More recent studies have further enhanced such models using deeper residual structures, multiscale convolutions, and related feature enhancement strategies in order to capture subtle waveform variations more effectively [16]. These developments show that convolution-based models remain a highly effective backbone for general ECG classification, especially in single-beat tasks.

In addition to pure convolutional models, hybrid architectures that combine convolutional and temporal modeling have also been widely investigated. Since ECG classification often benefits not only from local morphology but also from temporal dependencies among neighboring beats, CNN-LSTM combinations have become a common design choice. A representative recent example is the lightweight hybrid CNN-LSTM explainable model proposed by Alamatsaz *et al.* [18]. In their framework, CNN layers were used to extract local waveform features, LSTM was employed to model temporal dependencies, and SHAP was incorporated to improve interpretability. The significance of this work lies in its attempt to balance three different goals at once: accurate arrhythmia detection, temporal modeling of heartbeat sequences, and a degree of explainability. At the same time, however, it also illustrates a typical limitation of hybrid deep architectures: once convolution, recurrence, and explanation modules are stacked together, both structural complexity and inference cost increase noticeably.

Another active direction in recent years is the use of Transformer-based architectures. Compared with CNNs and RNNs, Transformers are designed to model long-range dependencies more explicitly through attention mechanisms [16]. ECGTransForm, proposed by El-Ghaish and Eldele, is a representative example in this direction [19]. Their framework integrates a bidirectional Transformer with multiscale convolutions and a channel recalibration module, and further introduces a context-aware loss to alleviate class imbalance. This work is important because it shows that Transformer-based sequence modeling is no longer limited to natural language processing or generic time-series analysis, but is being systematically adapted to ECG arrhythmia classification. However, the same study also highlights a practical limitation: to achieve stronger global dependency modeling, such frameworks usually require additional modules and increased computational cost, making them attractive for high-performance general classification but less naturally compatible with strict resource constraints.

Overall, the development of general ECG classification methods reveals a relatively clear methodological trajectory. Traditional methods emphasize handcrafted features and lightweight classifiers. Their strengths are structural transparency and interpretability, but their performance is often limited by feature engineering [17, 15]. Deep learning methods emphasize automatic representation learning and end-to-end optimization. Their strengths lie in stronger classification performance and more flexible modeling of complex patterns, but they usually come with larger input size and higher model complexity [15, 16, 19]. In other words, most general ECG classification methods have primarily addressed the question of how to achieve better classification performance on public datasets, rather than how to make such methods suitable for resource-constrained deployment. This observation naturally motivates the next subsection.

2.4 Low-Power and On-Chip ECG Classification

When the application scenario of ECG classification shifts from general offline analysis to wearable devices, edge nodes, and on-chip systems, the research focus changes accordingly. In server or desktop environments, models are usually optimized primarily for classification accuracy, while input length, model size, and inference cost are not necessarily the dominant constraints. In resource-constrained scenarios, however, computation capability, on-chip memory, and energy budget are all limited. As a result, the practical value of a method depends not only on classification performance, but also on whether it can operate reliably under strict resource constraints [16]. From this perspective, low-power ECG classification is not simply an extension of general ECG classification, but rather a reformulation of the problem under tighter system-level constraints.

Existing studies on low-power and on-chip ECG classification can be broadly divided into two main directions. The first aims to reduce the amount of data entering the classifier, thereby suppressing redundancy at the input representation level. The second aims to reduce model complexity through lightweight network design, model compression, and hardware-friendly implementation [20, 21, 22]. The former mainly addresses the problem of excessive input data, whereas the latter mainly addresses the problem of excessive model complexity.

Regarding input reduction, Hua *et al.* proposed a compressed-domain ECG classification method in which preprocessing, compression, and classification are integrated into a unified pipeline instead of reconstructing the full ECG prior to classification [20]. The significance of this work lies in showing that, in wearable scenarios, reducing input length itself can be an effective way to reduce overall system cost. The value of input-side optimization is not limited to saving storage and transmission cost; more importantly, it reduces the amount of data that subsequent models need to process. At the same time, however, this line of research also faces a clear challenge: once the input is compressed, sufficient discriminative information must still be preserved, otherwise classification performance may degrade significantly. Therefore, the essence of this direction is not simply to make the data smaller, but to seek a more compact yet still discriminative signal representation.

A related idea does not necessarily rely on explicit compression, but instead redesigns the input organization itself so that the signal representation is more suitable for resource-constrained implementation. In this view, the signal format presented to the classifier should not be regarded as fixed, and input design itself becomes part of system optimization. This perspective is evident in hardware-oriented ECG classification studies. For example, the configurable hardware-efficient ECG classification inference engine proposed by Zhang *et al.* mainly focuses on efficient inference, but it also demonstrates an important point: in resource-constrained scenarios, system efficiency is determined not only by the classifier itself, but also by how front-end data are organized, buffered, and fed into the model [21]. This observation suggests that input design is not merely a signal-processing issue, but an integral part of low-power system design.

Alongside input reduction, another major direction is model-side lightweighting. Zhang *et al.* already showed that classifier design for resource-constrained platforms cannot be treated as a purely algorithmic problem, but must be considered jointly with hardware execution [21]. On this basis, many studies further attempt to reduce model parameters and inference cost through network compression. A representative example is the pruned lightweight neural network proposed by Liu *et al.*, where parameter count and FLOPs were significantly reduced through pruning while competitive classification performance was maintained [22]. The value of such work lies in demonstrating that considerable structural redundancy does exist in ECG classification networks, which makes model compression practically feasible.

However, model compression also has a typical limitation. Even when the network itself becomes smaller, the input often remains a relatively complete ECG waveform, which means that the overall system burden may not decrease proportionally [22]. In other words, pruning and structural compression mainly solve the problem of model size, but do not necessarily solve the problem of excessive input data. From a system-level point of view, this means that model compression, while important, is not sufficient on its own to constitute a complete low-power solution.

More recent studies have gone a step further by treating extremely small models as a design target in themselves, rather than merely as compressed versions of large models. A representative example is the tiny Transformer proposed by Busia *et al.* [23]. Their model achieves low-power arrhythmia classification with only a very small number of parameters and is successfully deployed on a microcontroller with very low inference energy. The importance of this work lies in showing that even Transformer architectures, which are often considered computationally heavy, can be redesigned for microcontroller-scale deployment. At the same time, it also indicates that, in low-power ECG classification, truly valuable innovation may come not from increasing architectural complexity, but from preserving sufficient discriminative capability within an extremely small computational budget.

Similarly, the tiny matched-filter-based CNN proposed by Farag represents another lightweight convolutional route [24]. This work not only emphasizes a small model size, but also validates the method under the inter-patient setting. This is particularly important because a practically meaningful lightweight ECG classifier should not only work under relaxed conditions, but should also maintain some degree of generalization when encountering unseen subjects. Such studies show that compact models are not necessarily limited to oversimplified tasks, although they also make clear that achieving both lightweight design and robust generalization is inherently challenging.

Beyond algorithm- and model-level optimization, some recent studies have started to place ECG classification back into the context of complete systems. A representative example is the ECG-on-Chip architecture proposed by Xu *et al.* [25]. Their system not only includes front-end acquisition modules, but also incorporates arrhythmia-aware data transmission, meaning that the classification output directly affects whether subsequent data transmission is required. The significance of this work lies in the fact that ECG classification is no longer treated as an isolated algorithmic block, but is explicitly linked to sensing, communication, and system-level energy allocation. In wearable scenarios, this is highly meaningful, because classification does not merely generate a label; it may also determine subsequent system behavior.

Taken together, these studies indicate that the research priorities of low-power and on-chip ECG classification differ substantially from those of general offline classification. On the one hand, researchers are increasingly concerned with whether a classifier can operate on real hardware platforms, rather than simply whether it can achieve high accuracy on PCs or GPUs [23, 24, 25]. On the other hand, the literature has gradually recognized that system cost is determined not only by model size, but also by input data volume, signal representation, and the coordination among sensing, classification, and communication [20, 21, 25]. Therefore, low-power ECG classification should not be understood simply as shrinking a high-accuracy model, but rather as a joint optimization problem involving input representation, classifier architecture, and system processing flow.

Despite the substantial progress made in this field, several limitations remain. First, many lightweight solutions still rely on task simplification, such as reducing the number of classes or adopting easier evaluation settings, which makes them less directly applicable to more complete heartbeat classification problems

[24, 23]. Second, many studies optimize only one side of the problem: either they mainly compress the model while leaving the input unchanged, or they mainly reduce the input burden without synchronously redesigning the classifier [20, 22]. Third, even when hardware deployment results are reported, many studies mainly demonstrate that deployment is feasible, rather than explaining why a particular combination of input representation and classifier structure is especially suitable for resource-constrained scenarios [21, 25]. In this sense, although the literature has already provided many important explorations, a truly systematic answer to the question of how to reduce both input redundancy and model complexity at the same time is still lacking [20, 22, 25].

Based on the above analysis, it can be seen that although low-power and on-chip ECG classification has made substantial progress, its core challenge remains unresolved: high-performance methods often depend on larger inputs and more complex models, whereas truly lightweight solutions often involve task simplification or information loss. Therefore, it remains worthwhile to investigate how to simultaneously reduce the amount of data entering the classifier and control the complexity of the classifier itself, while still preserving meaningful arrhythmia recognition capability. This is exactly the motivation for the subsequent chapters: Chapter 3 focuses on reducing redundancy through a more compact ECG representation, while Chapter 4 further explores a classification framework that is better suited to resource-constrained deployment.

3 Level-Crossing-Based ECG Representation and Feature Optimization for Single-Beat Classification

Chapter 2 has shown that existing ECG classification methods face two major challenges in resource-constrained scenarios. First, the input data volume of raw ECG waveforms is often large. Second, even after preliminary compression, directly feeding relatively high-dimensional data into a neural network may still result in a non-negligible model size and computational burden. Therefore, for low-power ECG classification, reducing the overall system cost cannot rely solely on compressing the classifier itself, but must instead address redundancy at both the signal representation level and the feature input level.

To address this problem, we develops a two-stage framework for low-power single-beat ECG classification. First, a level-crossing (LC) event-driven representation is adopted to express the original ECG signal in a more compact form, thereby reducing the amount of data entering the subsequent processing pipeline, while also offering a more suitable alternative to computationally heavier approaches such as wavelet transforms, frequency-domain analysis, and other complex signal transformations. Then, based on the LC representation, baseline features are constructed and their limitations are analyzed through preliminary classification results. On this basis, rhythm-related, enhanced morphological, and LC-dynamic features are further introduced to progressively expand and optimize the feature set. Finally, by comparing the proposed method with a baseline approach and with representative results reported in the literature, we demonstrate that the proposed framework can effectively reduce input data volume and computational complexity while still maintaining competitive classification performance.

3.1 Level-Crossing-Based ECG Representation

In low-power wearable ECG classification, the way the input data are represented is itself a fundamental design issue. Conventional ECG acquisition usually relies on uniform sampling, where signal values are recorded continuously at a fixed sampling frequency. For example, in the MIT-BIH Arrhythmia Database, ECG signals are sampled at 360 Hz, which means that 360 samples are generated every second. For offline analysis, this representation is natural because it preserves the temporal continuity of the waveform and is convenient for subsequent filtering, frequency-domain analysis, and deep learning models. However, in resource-constrained scenarios, uniform sampling is not necessarily the most efficient representation. The fundamental reason is that the information carried by ECG signals is not uniformly distributed over time.

Around the QRS complex, the waveform changes rapidly and contains rich local morphological information, whereas in relatively flat segments, such as the baseline or slowly varying ST–T regions, the differences between neighboring samples are much smaller. If the same sampling density is applied over the entire signal, many samples are effectively spent on repeatedly describing low-variation regions, which introduces considerable redundancy.

This redundancy further affects the subsequent low-power classification pipeline. A larger number of samples means that more data need to be acquired, buffered, and transmitted at the front end. Moreover, if these raw waveform points are directly fed into the classifier, the input dimensionality of the network also increases, which in turn raises the model size and inference cost. In other words, under resource-constrained conditions, the problem is not only that the classifier itself may be too complex, but also that the data entering the classifier may already be excessive. Therefore, reducing data redundancy at the signal representation level is not merely an auxiliary optimization, but a prerequisite for low-power ECG classification.

To address this issue, this research adopts a level-crossing (LC) event-driven representation as the front-end signal representation of ECG. Figure 8 shows the typical ECG waveform and its Level-crossing representation. Unlike conventional uniform sampling, LC representation does not record signal values at fixed time intervals. Instead, it generates an event only when the signal crosses a predefined quantization level. In other words, it does not "sample once every fixed time interval," but rather "records an event only when the waveform changes sufficiently." This mechanism changes the sampling process from time-driven to change-driven. In regions where the waveform changes rapidly, such as around the QRS complex, the signal crosses multiple quantization levels in succession, and events therefore become dense. In regions where the waveform changes slowly, the number of level crossings is much smaller, and the event density is correspondingly reduced. As a result, the sampling resources are no longer uniformly distributed over the time axis, but are adaptively concentrated on the more informative portions of the signal. From the perspective of representation, the advantage of LC lies not in simply reducing the number of samples, but in reorganizing the sampling process according to the waveform dynamics, so that the major morphological variations are preserved while redundant records in low-variation regions are reduced.

This property is particularly suitable for ECG signals. In beat-level classification, discriminative information is usually not uniformly distributed across the entire heartbeat. Instead, it is more concentrated in rapidly varying local regions, such as the rising and falling slopes of the QRS complex, the slope changes around the R peak, and the relative positional relationships between the QRS complex and its neighboring

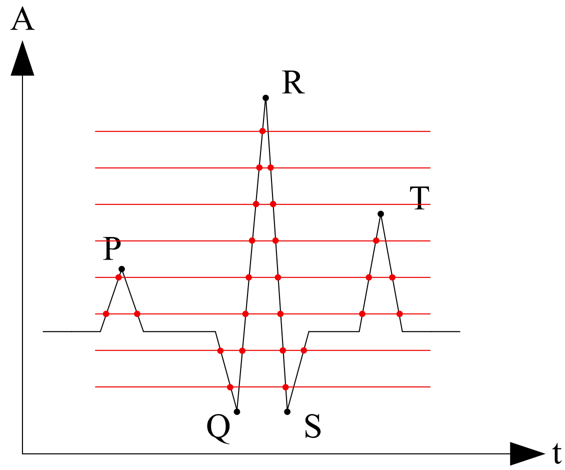

Figure 8: Typical ECG Waveform and its Level-crossing Representation

waveform components. In contrast, longer flat segments may still carry clinical meaning, but they do not necessarily need to be represented with the same sampling density. LC representation naturally matches this characteristic of ECG: it preserves a higher event density in fast-changing regions to describe local morphology, while it reduces the number of events in flatter regions to compress the data. Therefore, LC does not indiscriminately "remove samples," but rather re-encodes the input in a manner that is more consistent with the intrinsic variation pattern of ECG waveforms. This is why it can achieve data compression while still retaining the main structural information needed for subsequent classification.

There is also a more direct methodological reason for adopting LC representation in our research, namely, that it is fully aligned with the objective of first reducing the input data volume and then lowering the computational complexity of subsequent classification. Although wavelet transforms, frequency-domain analysis, and other multi-scale time–frequency methods can also extract useful information from ECG signals, such methods are usually built upon relatively complete uniformly sampled waveforms and require additional transformation operations and intermediate representations. For offline analysis or high-performance platforms, these methods are certainly reasonable and widely used. However, in the low-power scenario considered in this work, the extra computation and intermediate storage would further increase the implementation cost. From a system-level perspective, this means that even if the extracted features are discriminative, the front end still has to preserve a relatively large amount of raw data before those features can be computed. In contrast, LC representation acts directly on the sampling process itself and reduces the amount of data before the signal enters the subsequent processing pipeline. Therefore, it better matches the methodological chain of our research. In other words, LC is not introduced here as an additional signal analysis tool, but as the fundamental front-end representation of the entire low-power classification framework, whose role is to first address the problem of excessive input data volume.

In LC representation, parameter configuration directly affects both compression performance and waveform fidelity, and therefore a balance must be achieved between data reduction and information preservation. For the classification task considered in our work, it is not necessary to reconstruct the original uniformly sampled waveform with perfect accuracy, but it is necessary to preserve the key information related to beat morphology and rhythm as much as possible. Therefore, the LC parameters cannot be selected solely to maximize compression ratio, nor solely to minimize distortion. Instead, they should be determined by considering the requirements of the subsequent classification task. The main LC parameters considered in this work include the input full-scale voltage range, the quantization resolution, and the hysteresis/debouncing parameter. The input full-scale voltage range determines the overall amplitude interval covered by the LC representation, the quantization resolution determines how many discrete levels this interval is divided into, and the debouncing parameter is used to suppress repeated crossing events caused by local noise and small oscillations. Together, these parameters determine the density of the LC event stream, the amplitude representation precision, and the sensitivity to noise.

First, the input full-scale voltage range determines the amplitude interval covered by the LC representation. If the input full-scale voltage range is too small, large-amplitude waveform variations may exceed the quantization interval and thus be clipped. If the input full-scale voltage range is too large, then for a fixed quantization bit width, the spacing between adjacent quantization levels becomes larger, which reduces amplitude resolution. In other words, a larger input full-scale voltage range provides stronger coverage capability but coarser amplitude representation, whereas a smaller input full-scale voltage range improves local resolution but may fail to cover the full signal variation. Therefore, choosing the input full-scale voltage range is essentially a trade-off between coverage and amplitude resolution. In this work, a input full-scale voltage range of 10 mV is adopted. On the one hand, this range is sufficient to cover the major amplitude variations of ECG waveforms and helps avoid obvious clipping of common beat morphologies during quantization. On the other hand, 10 mV is also a commonly adopted input full-scale voltage range in prior LC-ECG studies, which facilitates consistent comparison with existing work [26].

Second, the quantization resolution determines how many quantization levels are used within the input full-scale voltage range. If the quantization bit width is denoted by b, then the total number of levels is 2^b, and the spacing between adjacent quantization levels can be expressed as

$$\Delta = \frac{V_{\text{range}}}{2^b}, \tag{6}$$

where V_{range} is the selected input full-scale voltage range and Δ is the amplitude interval between adjacent quantization levels. This parameter directly determines how sensitive the LC representation is to waveform variations. When the quantization bit width is low, the level spacing is large, and only relatively obvious amplitude changes can trigger crossing events. As a result, the number of events is small and the compression ratio is high, but some subtle waveform variations may be ignored. For example, a low-amplitude P wave may fail to generate enough crossing events under coarse quantization, which may lead to missed detection or insufficient morphological description. In contrast, when the quantization bit width is high, the level spacing becomes smaller, and the system responds to finer waveform variations, thereby describing ECG morphology more precisely. However, the number of crossing events also increases significantly, and the compression effect is weakened. If the level spacing becomes too small, even minor fluctuations may trigger events frequently, and in some local segments the number of events may even exceed the number of uniformly sampled points, which defeats the purpose of compression. Therefore, the quantization resolution determines the balance between compactness and detail preservation.

To quantitatively evaluate this trade-off, this work uses the compression ratio (CR) and the signal-to-distortion ratio (SDR) as two comparison metrics. The compression ratio describes how much the LC representation reduces the amount of data relative to the original uniformly sampled signal, and is defined as

$$CR = \frac{N_{\text{uniform}}}{N_{\text{LC}}}, \tag{7}$$

where N_{uniform} denotes the number of samples in the original uniformly sampled ECG and N_{LC} denotes the number of LC events. A larger CR indicates stronger compression. At the same time, it is also necessary to measure the distortion introduced by LC representation. In our work, the SDR is defined as

$$\text{SDR} = 10\log_{10}\left(\overline{(x-\bar{x})^2}\right) - 10\log_{10}\left(\overline{(x-\hat{x})^2}\right), \tag{8}$$

where x is the original uniformly sampled signal and $\hat{x}$ is the approximate waveform reconstructed from the LC event stream. A higher SDR indicates lower distortion and better waveform fidelity. As shown in Table 2, when the quantization resolution increases from 5-bit to 8-bit, the compression ratio decreases from 32.72 to 2.36, while the SDR increases from 8.94 dB to 24.78 dB. This means that a higher quantization resolution indeed improves waveform fidelity, but at the cost of significantly weaker compression. In particular, although 5-bit and 6-bit provide relatively high compression ratios, their SDR values are low, indicating substantial waveform distortion. On the other hand, although 8-bit gives the highest fidelity, the number of events increases so much that the compression advantage becomes marginal. Based on [27],"Good" quality signals must follow the following criteria:

$$SDR \geq 21dB. \tag{9}$$

Considering both compactness and waveform fidelity, 7-bit provides a more balanced trade-off and is therefore selected as the quantization resolution for the subsequent experiments.

Table 2: Compression Ratio and SDR under Different LC Resolutions.

Resolution (bit)	5	6	7	8
CR	32.72	12.41	5.37	2.36
SDR (dB)	8.94	15.23	19.86	24.78

Besides the input full-scale voltage range and quantization resolution, the debouncing parameter also has a significant impact on the quality of the LC event stream. In practical ECG signals, when the waveform is close to a quantization level, local noise and small oscillations may cause repeated back-and-forth crossings between adjacent levels, thereby generating many meaningless redundant events. Figure 9 shows repeated noise-induced crossings near a quantization level. This phenomenon substantially increases the event count and may further contaminate crossing-count-based features. To suppress this effect, a hysteresis-based debouncing mechanism is introduced in the LC representation. Its key idea is not simply to require the signal to remain stable for some duration, but rather to require that, after the waveform changes direction, a sufficient number of quantization levels must be crossed in the new direction before a new event is accepted. If the number of crossed levels is insufficient, the change is treated as a local oscillation and no new event is recorded. Let the debouncing parameter be denoted by k. Then k can be interpreted as the minimum number of levels that must be crossed after a direction reversal for a new reverse crossing to be accepted.

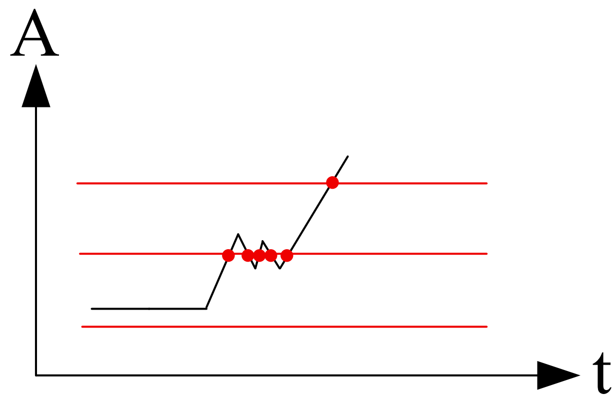

Figure 9: Illustration of Repeated Noise-induced Crossings near a Quantization Level.

Under this mechanism, the minimum effective amplitude change can be expressed as

$$\Delta_{\text{min}} = k\Delta. \tag{10}$$

For example, when $\Delta = 0.078125$ mV and $k = 4$, a reverse event is accepted only when the cumulative amplitude change after a direction reversal reaches approximately 0.3125 mV. If the waveform only fluctuates slightly around a quantization boundary, such small back-and-forth changes will not continuously generate new events. In contrast, a true QRS complex usually exhibits much faster and larger amplitude changes, so even with debouncing it can still cross multiple quantization levels and be effectively preserved. Hence, the role of k is essentially to balance the suppression of redundant noise-induced oscillatory crossings against the preservation of meaningful rapid waveform transitions.

For the specific choice of k, this work refers to the parameter analysis reported by Ravanshad *et al.* in their LC-based QRS detection study [28]. They showed that when $k = 1$ LSB, the LC output is highly sensitive to noise around the crossing point and may generate multiple redundant crossings. By comparing different parameter combinations on the MIT-BIH Arrhythmia Database, they identified $k = 4$ as a preferred design point because it provides a favorable balance among noise suppression, sampling-rate reduction, and preservation of meaningful waveform information. Following this analysis, and in conjunction with the 7-bit quantization resolution adopted in this work, the debouncing parameter is set to $k = 4$. This setting helps suppress redundant oscillatory crossings while retaining the major morphological variations of ECG waveforms.

In summary, we adopt a input full-scale voltage range of 10 mV, a 7-bit quantization resolution, and a debouncing parameter of $k = 4$ as the unified LC configuration for the subsequent experiments. This parameter combination is not chosen merely to maximize the compression ratio, but rather to achieve a practical trade-off among signal coverage, amplitude resolution, event-count control, and downstream classification usability. The results show that this configuration can significantly reduce the amount of raw sampled data while still preserving the major morphological characteristics of ECG waveforms, and is therefore suitable as the basis for the subsequent feature construction and classification study in our research.

It should be further noted, however, that although LC representation has already removed a substantial amount of redundancy from the original uniformly sampled waveform, this does not mean that the classifier input has automatically become sufficiently compact. According to [28], under the selected LC setting with $k = 4$, the effective sampling rate after LC conversion is approximately 67 Hz. If the heart rate is assumed to be 75 bpm, then the average duration of one heartbeat is about 0.8 s, which corresponds to roughly 54 LC samples per beat. In other words, even after LC compression, directly feeding the LC sequence into a neural network would still require the classifier to process on the order of several dozen input samples for each heartbeat.

By contrast, the final representation adopted in this work compresses each heartbeat into only 13 features. This means that the input dimensionality of the classifier is reduced from about 54 LC samples to 13 compact discriminative features, leading to a substantial reduction in the input size of the neural network. Therefore, LC mainly addresses the redundancy in raw ECG sampling, but it does not by itself guarantee that the classifier input is already optimal. Further feature extraction is still necessary in order to transform the LC sequence into a more compact and classification-oriented representation. This is exactly the problem addressed in the next section.

3.2 Baseline Feature Construction and Preliminary Analysis

After establishing the LC representation, the next step is not to directly feed the LC event sequence into the classifier, but to further construct a compact and discriminative heartbeat-level representation. Although the LC representation has already removed a substantial amount of redundancy from the original uniformly sampled ECG at the signal-representation level, the LC event stream itself still preserves considerable temporal and amplitude-related information. If it is directly unfolded as the input to a classifier, the input dimensionality may still remain relatively high, and the size and computational complexity of the subsequent model would be difficult to reduce further. Therefore, in this work, feature extraction is introduced on top of the LC representation so as to achieve a further transition from raw-data compression to classifier-input compression.

In this process, constructing a set of baseline features is necessary. The purpose of the baseline is not to immediately achieve the optimal classification performance, but rather to provide a structured and interpretable reference point for the subsequent feature-expansion and feature-optimization stages. For beat-level ECG classification, the most basic and physiologically meaningful information still comes from the geometric and morphological relationships among the main heartbeat fiducial points. Compared with directly introducing large-scale deep features, frequency-domain features, or more complex transformed descriptors, such features maintain a relatively direct correspondence with conventional ECG interpretation, which makes it easier to analyze the source of inter-class differences. At the same time, they are low-dimensional and structurally clear, making them suitable as a baseline for later expansion and selection. Consequently, the first step of LC-based feature construction in this work is to establish a compact and interpretable baseline centered on the fundamental morphology of individual heartbeats.

Before formally defining the baseline feature vector, however, it is necessary to clarify how the required heartbeat fiducial points are obtained from the LC event stream. Unlike uniformly sampled ECG, the LC signal is no longer represented as a sequence of waveform amplitudes sampled at regular time intervals. Instead, it is expressed as an unevenly spaced sampling points of rise and fall crossing events generated when the signal crosses predefined quantization levels. As a result, the basic information unit in LC ECG is no longer "the signal amplitude at a given sampled moment," but rather "the occurrence of an upward or downward crossing event at a specific time." This means that the structure of an individual heartbeat is not explicitly presented as a continuous waveform, but is instead embedded in the direction transitions of events, the intervals between events, and the local density of the event stream. Therefore, if discriminative heartbeat-level features are to be constructed in the LC domain, the first prerequisite is to perform heartbeat segmentation and fiducial-point localization directly on the LC event stream.

To this end, this work adopts an event-driven heartbeat-mapping procedure that converts the continuous LC ECG stream into a structured beat-by-beat representation. The core idea is not to reconstruct the full waveform, but rather to directly identify the most important local structures of each heartbeat in the event domain and use them as the basis for later feature construction. This design is motivated by three considerations. First, the most diagnostically relevant information in ECG classification is concentrated around a limited number of physiologically meaningful regions, especially the QRS complex, T wave, and P wave. Second, under the LC representation, the transitions among these structures are naturally reflected by the rise/fall event patterns themselves, which already provide a compact description of local morphology. Third, since the overall goal of this work is low-power and resource-constrained classification, the feature-extraction process should remain compact and implementation-friendly rather than reintroducing a heavy waveform-reconstruction or high-dimensional processing stage.

Figure 10 shows the flowchart of event-driven heartbeat segmentation and fiducial-point localization under the LC representation. In the proposed event-driven extraction process, the LC ECG is first treated as a time-ordered sequence of rise and fall events, and the detection logic is built around the QRS complex, which corresponds to the fastest local waveform variation. Since the region around the R peak usually exhibits the most rapid amplitude change and the densest event activity within a heartbeat, the event-driven localization process begins there. Specifically, the algorithm first detects a candidate rise-fall pattern. Once such a pattern is observed, it starts counting the subsequent consecutive fall events. If the number of consecutive fall events exceeds a predefined threshold, the local segment is treated as a valid RS region. Within this region, the first rise-fall transition is marked as the R peak, while the subsequent fall-rise transition is marked as the S point. In this way, the two most important structural points within the QRS complex can be identified directly from the LC event stream without relying on a reconstructed uniformly sampled waveform.

After the R and S points are determined, the algorithm further searches forward from the S point to locate the T point. In the LC domain, the S-T to T-wave rising portion is typically reflected by consecutive rise events. Therefore, starting from the detected S point, the algorithm scans forward through the rise events until a new fall event is encountered, and the corresponding turning location is identified as the T point. The rationale is that the T wave usually forms the next prominent local peak after the QRS complex, and

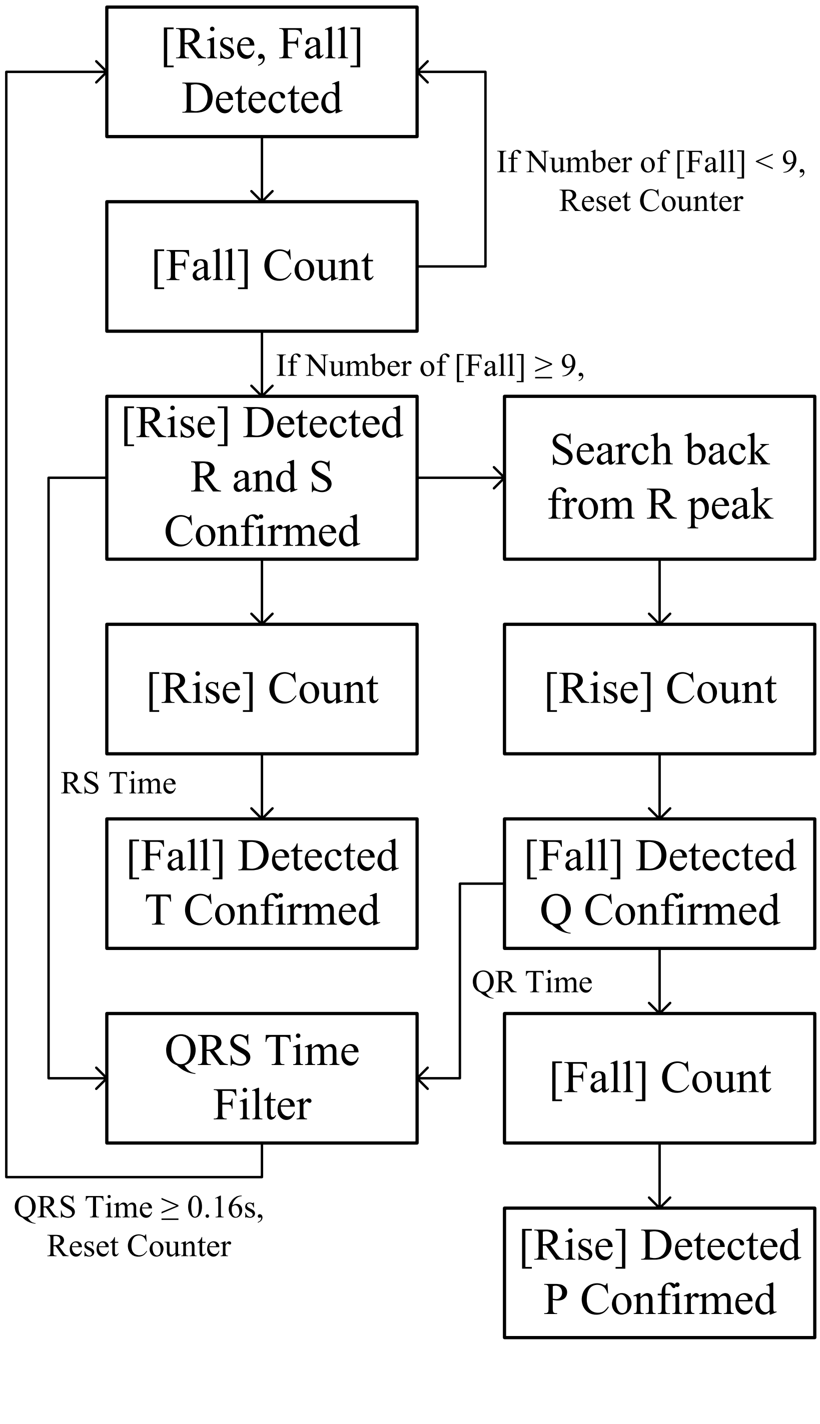

Figure 10: Event-driven Heartbeat Mapping for Baseline Feature Extraction

in the event-driven representation this peak naturally appears as a sustained rising trend followed by a fall transition. Accordingly, the T point can be recovered through this rise-to-fall directional change.

After R, S, and T have been identified, the algorithm further determines the Q and P points. Unlike the forward search used for T, the Q-point localization is performed by searching backward from the detected R peak. Since the Q wave usually corresponds to a local valley preceding the R peak, it can be represented in the LC domain by a characteristic rise-to-fall transition before the main R event. Based on this observation, the algorithm searches backward from the R peak and identifies the first valid rise-to-fall transition as the Q point. After that, the earlier portion of the event sequence is further examined to find the first fall-to-rise transition, which is then marked as the P point. Through this combination of forward and backward event-driven searches, the major heartbeat fiducial points P, Q, R, S, and T can all be localized in a unified manner directly from the LC event stream.

It should be noted that relying solely on local event patterns for R-peak detection may lead to false detections caused by large-amplitude T waves, local noise fluctuations, or abnormal crossing patterns. To reduce such errors, this work further introduces a QRS-duration filter as a physiological constraint. Specifically, once candidate Q, R, and S points are found, the durations of the QR and RS subsegments are combined to estimate the overall QRS duration. If this duration exceeds a predefined threshold, the candidate is regarded as more likely to correspond to a false detection caused by an abnormally large T wave or noise-induced event sequence rather than a true QRS complex, and the detection is discarded while the counters are reset. This additional step helps constrain the physiological plausibility of the detected fiducial points and improves the stability and consistency of the entire event-driven extraction procedure.

After the above heartbeat-segmentation and fiducial-point-localization procedure, the originally continuous LC ECG stream is transformed into a structured beat-level representation. For each heartbeat, the temporal positions and relative amplitudes of the P, Q, R, S, and T points can then be obtained. This step is critical because not only the baseline morphology used in the present section, but also the rhythm-related, enhanced morphological, and LC-dynamic features introduced later in Section 3.3, are all ultimately built upon this event-driven fiducial-point extraction process. Without this step, the later definitions of A_P, A_Q, A_R, A_S, A_T and the corresponding temporal intervals would appear to arise directly from the LC signal without a clearly described extraction path.

Based on the localized fiducial points, the baseline feature vector is then constructed. In beat-level ECG analysis, the P, Q, R, S, and T points and their mutual relationships form the most basic source of morphological information. The differences among heartbeat categories are often first reflected in these local structures, such as changes in QRS width, shifts in the relative position of the R peak, altered peak-valley amplitude relationships, and different temporal unfolding patterns along the heartbeat. Therefore, once these fiducial points can be reliably located in the LC representation, a low-dimensional feature set centered on local morphological relationships can be defined accordingly.

Specifically, the baseline features in our study are constructed from the amplitude relationships and temporal relationships among heartbeat fiducial points under the LC representation. The fiducial points considered here include the P, Q, R, S, and T positions, which jointly define the most basic morphological structure of a heartbeat. In the baseline design, two aspects are mainly considered: the amplitudes at the key points, which reflect the strength of local electrical activity, and the temporal intervals between neighboring fiducial points, which characterize the local structural expansion of the waveform along the time axis. Based on this idea, five amplitude features are defined as A_P, A_Q, A_R, A_S, A_T, which correspond to the amplitudes of the P wave, Q wave, R peak, S wave, and T wave, respectively. Specifically, A_R represents the peak amplitude of the QRS complex and is typically the most prominent feature in an ECG beat, while A_Q and A_S describe the local minima around the R peak. The amplitudes A_P and A_T capture the atrial depolarization and ventricular repolarization phases, respectively. These amplitude features reflect the morphological characteristics of a heartbeat and provide essential information for distinguishing different arrhythmia types. Another four temporal relational features are defined as

$$\Delta t_{PQ} = t_Q - t_P, \tag{11}$$

$$\Delta t_{QR} = t_R - t_Q, \tag{12}$$

$$\Delta t_{RS} = t_S - t_R, \tag{13}$$

$$\Delta t_{ST} = t_T - t_S. \tag{14}$$

Here, t_P, t_Q, t_R, t_S, and t_T denote the temporal positions of the P, Q, R, S, and T waves, respectively. These temporal features describe the local temporal structure of the ECG waveform. Specifically, Δt_{PQ} reflects the interval between atrial depolarization and the onset of ventricular depolarization, Δt_{QR} and Δt_{RS} characterize the internal timing of the QRS complex, and Δt_{ST} captures the transition from ventricular depolarization to repolarization. Compared with absolute timing information, these relative intervals are less sensitive to heart rate variations and therefore provide more robust features for classification. Accordingly, the baseline feature vector can be written as

$$\mathbf{F}_{\text{baseline}} = \{A_P, A_Q, A_R, A_S, A_T, \Delta t_{PQ}, \Delta t_{QR}, \Delta t_{RS}, \Delta t_{ST}\}. \tag{15}$$

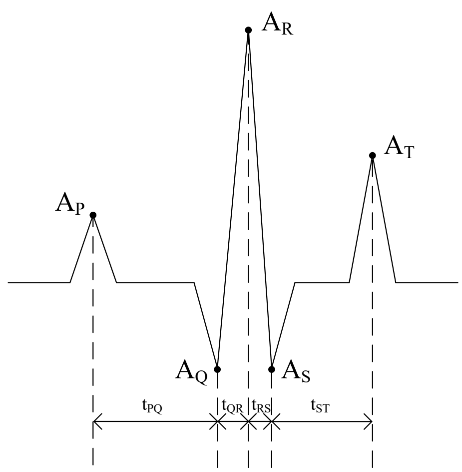

Figure 11: The 9 Baseline Features of a Typical ECG Waveform

This feature set has a total dimensionality of 9. These baseline features are shown in Figure 11. Overall, all features are derived from local intra-beat structures, which gives them clear physiological interpretability and also provides a low-dimensional and interpretable reference for subsequent feature expansion. It should be noted that, although Chapter 2 introduced the AAMI-recommended five-class framework, namely N, S, V, F, and Q, the experiments in this work adopt a four-class setting including only N, S, V, and F. The reason is that the Q class is closer to a residual or unclassified category, with relatively high heterogeneity and weak intra-class consistency, while the number of corresponding samples in the MIT-BIH Arrhythmia Database is also limited. Since this study focuses on LC representation and feature optimization, including such a weakly defined and highly heterogeneous category would make it more difficult to clearly evaluate the contribution of feature design to the classification of major arrhythmia types. Therefore, the present work concentrates on the four more common and better-defined heartbeat classes and excludes the Q class from the current task.

To obtain the classification results of the baseline features, experiments were conducted on the MIT-BIH

Arrhythmia Database under the above four-class setting, and the data were organized using the Conditional Data Grouping Scheme (CDGS) [29]. The purpose of CDGS is to ensure the effectiveness of the training data under class-imbalanced conditions and to reduce the instability caused by records containing only a very small number of abnormal beats. Specifically, CDGS first counts the distribution of heartbeat types in each record and groups the records accordingly. If the proportion of a certain non-N beat type in a record is lower than 1% of the total number of beats in that record, the record is regarded as having insufficient samples for that class and is not used for the training of the corresponding classification task. Following this principle, the dataset can be divided into several groups, such as (N,S), (N,V), (N,S,V), (N,V,F), and (N). Among them, the (N) group, which contains only normal beats, is not involved in the subsequent abnormal beat classification task. A key characteristic of CDGS is that it adopts a patient-specific data organization strategy, where both training and validation are carried out within the same record, thereby avoiding the interference caused by inter-patient identity differences.

For classification, a multilayer perceptron (MLP) is used as the baseline classifier. The MLP is a typical feedforward neural network consisting of an input layer, one or more hidden layers, and an output layer, and it is able to learn nonlinear mappings between input features and class labels. Since the baseline features used in this study are low-dimensional handcrafted descriptors with clear physiological meanings, a relatively simple MLP is sufficient and more appropriate as a baseline model, allowing the analysis to focus on the discriminative power of the features themselves rather than on the complexity of the classifier. Specifically, the adopted MLP has an input dimension of 9, corresponding to the 9 extracted baseline features of each heartbeat, one hidden layer with 32 neurons, and an output dimension of 4 corresponding to the four heartbeat classes N, S, V, and F. Figure 12 shows the Structure of the Baseline MLP Classifier. The training procedure is as follows. First, the 9-dimensional baseline feature vector is extracted for each heartbeat. Then, according to the CDGS principle, for each eligible record, only the classes satisfying the sample proportion criterion are retained, and the corresponding samples within that record are randomly divided into 70% for training and 30% for validation. For example, for Record 208 belonging to the (N,V,F) group, 70% of the N, V, and F beats are randomly selected as the training set, and the remaining 30% are used as the validation set. The same principle is applied to the other groups. After that, the 9-dimensional training features are fed into the MLP for supervised learning of the four heartbeat classes. Once training is completed, the remaining 30% validation samples from the same record are used to evaluate the model, and the performance is summarized by metrics such as Accuracy and Macro-F1.

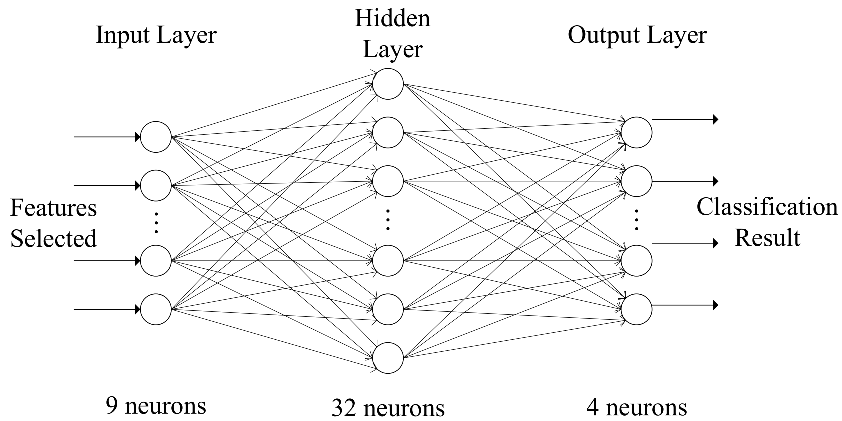

Figure 12: The Structure of the Baseline MLP Classifier

The overall classification performance of the baseline feature set is summarized in Table 3. The results show that, even with only these 9 handcrafted features derived from local intra-beat structures, the model can already achieve a reasonable classification performance, indicating that these features are able to capture discriminative differences among different heartbeat types to a certain extent. However, the Macro-F1 score also suggests that there is still room for improvement in terms of balanced performance across classes, which provides motivation for the subsequent feature expansion and optimization.

Table 3: Overall performance of baseline features

Metric	Value
Accuracy	0.889
Macro-F1	0.736

The class-wise performance is shown in Table 4.

Table 4: Class-wise baseline performance

Class	Precision	Recall	F1
N	0.964	0.972	0.968
S	0.598	0.472	0.528
V	0.865	0.847	0.856
F	0.341	0.218	0.266

In the baseline experiment, the classifier input is exactly the above 9-dimensional feature vector. The purpose is not to claim that these basic morphological features are already sufficient, but rather to evaluate whether the LC representation can support stable beat classification when only basic morphological information is used. The overall results show that the baseline features achieve an Accuracy of 0.889 and a Macro-F1 of 0.736 on the four-class task. This indicates that, even with only a small number of basic morphological features, beat classification under the LC representation is already feasible to a certain extent. In other words, the amplitudes of fiducial points and their local temporal relationships do provide useful discriminative information for the classifier. At the same time, however, this result also shows that relying solely on basic morphological features is still insufficient for higher-quality classification performance. In particular, the fact that the Macro-F1 remains noticeably lower than the overall Accuracy suggests that the classifier is still relatively weak in recognizing minority or difficult classes.

The class-wise results further clarify this point. For the N class, the baseline features achieve a Precision of 0.964, a Recall of 0.972, and an F1-score of 0.968, which is the best among the four classes. This is mainly because N beats dominate the dataset and also exhibit relatively stable local morphology, so even basic amplitude and temporal features are sufficient to form a consistent class representation. For the V class, the baseline also achieves relatively strong performance, with a Precision of 0.865, a Recall of 0.847, and an F1-score of 0.856. Compared with N beats, V beats are less frequent, but their local morphology is often more evidently distorted, such as widened QRS complexes, abnormal local amplitude patterns, or altered peak-valley relationships. Such differences can be partially captured by the baseline amplitude and temporal features.

In contrast, the performance on the S and F classes is significantly worse. For the S class, the Precision is 0.598, the Recall is 0.472, and the F1-score is only 0.528. For the F class, the Precision is 0.341, the Recall is 0.218, and the F1-score further drops to 0.266. The poor performance on the S class is mainly because supraventricular ectopic beats are often relatively similar to normal beats in local morphology, while their more distinctive characteristics are usually reflected in rhythm relationships, such as premature occurrence and changes in the RR-interval pattern with neighboring beats. Since the current baseline features only describe the internal structure of a single beat and do not explicitly include inter-beat rhythm information, the classifier tends to misclassify part of the S beats as N beats. As for the F class, its poor performance is even more understandable. Fusion beats are formed by the joint effect of normal and abnormal conduction, and their morphology often lies somewhere between normal and abnormal patterns. A small set of fiducial-point amplitudes and simple temporal intervals is insufficient to represent such mixed and ambiguous structures, making F the most difficult class for the baseline model.

These results show that, although the baseline features constitute a reasonable starting point, their limitations are also clear. The current baseline does not contain rhythm information, and therefore has limited capability in distinguishing categories that depend on beat-to-beat temporal relationships. Meanwhile, it uses only a small number of fiducial-point features. Although these features summarize the basic geometric structure of ECG beats, they cannot adequately describe finer-grained morphological information, such as detailed internal relationships within the QRS complex, slope-related differences, or more subtle amplitude relations across different waveform components. In addition, for classes such as F beats, which exhibit strong morphological ambiguity and blurred class boundaries, a 9-dimensional basic feature set is often insufficient to establish a stable decision boundary.

Therefore, the role of the baseline experiment in our research is not merely to provide a preliminary result, but more importantly to supply the analytical basis for subsequent feature expansion. It shows that, under the LC framework, the basic morphological relationships among beat fiducial points do provide useful classification information, but if higher classification performance is desired while still keeping the input dimensionality compact, additional information sources must be incorporated. Specifically, the missing information mainly lies in three aspects: rhythm-related information associated with neighboring beats, enhanced morphological features that can describe local structure differences in a finer manner, and dynamic features that directly reflect the variation characteristics of the LC event stream. Based on this analysis, the next section further introduces these additional information sources and performs a systematic expansion and optimization of the candidate feature set.

3.3 Feature Expansion

The baseline experiment in the previous section shows that, although basic intra-beat morphological features already enable a certain degree of beat classification under the LC representation, their discriminative capability still has clear limitations. In particular, for abnormal beat types with relatively ambiguous decision boundaries, such as S and F beats, the current baseline features are insufficient to establish stable classification boundaries. This indicates that a small set of fiducial-point amplitudes and simple temporal relations is still not enough to fully characterize the differences among abnormal beat types. Therefore, if the classification performance is to be further improved while keeping the input dimensionality as compact as possible, additional discriminative information must be introduced on top of the baseline.

From the limitations of the baseline, the currently missing information can be summarized into three aspects. First, the baseline does not include rhythm information across neighboring beats, and therefore it is weak in representing arrhythmia types whose discrimination depends on beat-to-beat temporal relationships. Second, the baseline description of morphology is still rather coarse. Although it includes amplitudes at key points and several local temporal intervals, it does not fully capture finer structural variations within the QRS complex or among different waveform components. Third, although the baseline is constructed on top of the LC representation, it does not directly exploit the dynamic variation information carried by the LC event stream itself, and therefore it does not fully utilize the additional representational advantage of LC over conventional uniformly sampled waveforms. Based on this analysis, this section further supplements rhythm information, enhanced morphological information, and LC dynamic information, and then uses them to build a more complete candidate feature pool.

To compensate for the lack of rhythm information in the baseline, we first introduce features related to the temporal relationships between neighboring beats. For supraventricular ectopic beats, the local morphology of the current beat may still be relatively similar to that of normal beats, while their more distinctive differences are often reflected in premature occurrence and in the variation of RR intervals before and after the beat. Therefore, relying only on the internal morphology of a single beat can easily lead to confusion between such beats and normal beats. To capture this beat-to-beat temporal relationship, we further consider the intervals between the current beat and the previous beat, the intervals between the current beat and the next beat, as well as the difference between these intervals and their local average relationship.

Specifically, four rhythm-related features are added on top of the baseline: the pre-RR interval, the post-RR interval, the RR difference, and the local mean RR interval. Let the R-peak time of the n-th beat be denoted by $t_R^{(n)}$. Then the pre-RR interval is defined as

$$RR_{\text{pre}}^{(n)} = t_R^{(n)} - t_R^{(n-1)}, \tag{16}$$

and the post-RR interval is defined as

$$RR_{\text{post}}^{(n)} = t_R^{(n+1)} - t_R^{(n)}. \tag{17}$$

The local RR difference is then written as

$$\Delta RR^{(n)} = RR_{\text{post}}^{(n)} - RR_{\text{pre}}^{(n)}, \tag{18}$$

and the local mean RR interval is defined as the average of the previous eight RR intervals:

$$RR_{\text{mean}}^{(n)} = \frac{1}{8} \sum_{i=n-8}^{n-1} RR^{(i)}, \tag{19}$$

where $RR^{(i)}$ denotes the i-th RR interval. This feature characterizes the local rhythm background preceding the current beat, so that the classifier can not only observe the instantaneous temporal relation between the current beat and its neighboring beats, but also refer to the overall rhythm context in which the beat occurs. By introducing these rhythm features, the classifier no longer relies solely on the local morphology of the current beat, but can also take into account the relative temporal position of the beat within the surrounding rhythm, which is especially important for recognizing rhythm-related abnormalities such as S beats.

In addition to rhythm information, we further enhance the description of local morphology. The baseline only uses a small number of absolute amplitudes at key points and simple temporal relations. Although these features can outline the most basic geometric structure of a beat, they are still insufficient to describe finer local differences. In practice, the features that truly distinguish different abnormal beats are not always the absolute amplitude of a specific point, but more often the relative relationships among different waveform components, such as amplitude differences between peaks and valleys, local widening or narrowing of certain segments, and relative structural relations among different parts of the QRS complex. Based on this consideration, a set of enhanced morphological features is constructed on top of the baseline to describe finer local structural differences among ECG fiducial points.

One direct category of enhanced morphological features is the amplitude-difference feature. Compared with the absolute amplitude features in the baseline, amplitude-difference features focus more on the relative relationships among different waveform structures, and are also more effective in reducing the influence of overall amplitude-scale differences across subjects. Specifically, several typical amplitude-difference relationships are considered, such as

$$D_{QR} = A_R - A_Q, \tag{20}$$

$$D_{RS} = A_R - A_S, \tag{21}$$

$$D_{ST} = A_T - A_S, \tag{22}$$

$$D_{PQ} = A_Q - A_P. \tag{23}$$

These features reflect the relative amplitude relationships between local peaks and valleys from different perspectives. For example, D_{QR} and D_{RS} mainly characterize the internal amplitude differences within the QRS complex, whereas D_{ST} and D_{PQ} complement the relative structural information in the S–T and P–Q regions, respectively.

On this basis, an additional temporal morphological feature is further introduced to describe the QRS structure on a more global time scale. Although the baseline already uses the local temporal relationships from Q to R and from R to S to describe the internal structure of the QRS complex, for some abnormal beat types the duration of the entire QRS complex is more discriminative than the duration of an individual subsegment. Therefore, the QRS duration is defined as

$$QRS_{\text{dur}} = t_S - t_Q, \tag{24}$$

which reflects whether the QRS complex is widened from a more global perspective and is particularly informative for ventricular abnormal beats.

Besides rhythm and enhanced morphology, we further introduce dynamic features directly derived from the LC event stream itself. This is one of the most important parts of the current feature expansion. Although the baseline and enhanced morphological features are both built upon the LC representation, they still mainly revolve around conventional waveform fiducial points, and do not directly exploit the dynamic variation information inherently contained in the LC event sequence. In fact, one important characteristic of LC representation, compared with uniform sampling, is that it directly reflects the variation intensity of the waveform in different regions through the event distribution itself. Therefore, in addition to conventional waveform key points, the LC event stream can also serve as an independent source of information.

The first dynamic feature introduced in this work is the crossing density. Its basic idea is to count the number of valid crossing events within a given time window and normalize it into an event density per unit time. Let N_{cross} denote the number of valid crossing events within a selected window, and let T_{win} denote the corresponding window length. Then the crossing density is defined as

$$D_{\text{cross}} = \frac{N_{\text{cross}}}{T_{\text{win}}}. \tag{25}$$

To focus this feature on the most discriminative local region, the counting window is centered around the R peak, namely from 60 ms before the R peak to 100 ms after the R peak, corresponding to a total window length of 0.16 s. Therefore, in this work the feature can be further written as

$$D_{\text{cross}} = \frac{N_{\text{cross}}^{QRS}}{0.16}. \tag{26}$$

This feature reflects the local activity level of waveform variation around the QRS region. For the same time-window length, a denser event distribution indicates more intense local waveform variation, whereas a sparser event distribution implies relatively smoother local changes.

The second dynamic feature is the maximum slope. In the LC representation, the local slope between two adjacent events can be approximated by the amplitude change divided by the time interval between them. Let the quantized amplitude and time stamp of the i-th LC event be denoted by V_i and t_i, respectively. Then the local slope between two adjacent events is defined as

$$S_i = \frac{V_{i+1} - V_i}{t_{i+1} - t_i}. \tag{27}$$

Based on this, the maximum absolute slope within the selected QRS-related region is defined as

$$S_{\max} = \max_i |S_i|. \tag{28}$$

Since the amplitude change between neighboring LC events is determined by the quantization step ΔV_{LC}, one may further write

$$V_{i+1} - V_i = \pm \Delta V_{LC}, \tag{29}$$

and therefore

$$S_i = \frac{\Delta V_{LC}}{\Delta t_i}, \tag{30}$$

where $\Delta t_i = t_{i+1} - t_i$. This shows that, in the LC domain, the slope feature is fundamentally related to the inter-event interval. Nevertheless, $S_{\max}$ emphasizes the fastest local transition rather than the overall event density, and therefore provides a distinct description from D_{cross}.

The third dynamic feature is the QRS area. Unlike the previous two features, which mainly characterize event density and local variation speed, the QRS area focuses on the joint effect of waveform amplitude and duration over the entire QRS complex. In the continuous-time form, it is defined as

$$A_{QRS} = \int_{t_Q}^{t_S} |x(t)|\, dt, \tag{31}$$

where $x(t)$ denotes the ECG waveform and t_Q and t_S indicate the beginning and end of the QRS complex, respectively. Under the LC representation, this integral is approximated by a trapezoidal summation over adjacent LC events:

$$A_{QRS} \approx \sum_{i \in QRS} \left| \frac{V_i + V_{i+1}}{2} \right| (t_{i+1} - t_i). \tag{32}$$

This feature jointly contains information about both amplitude and duration. For some abnormal beats, even when the local peak values or individual temporal relations are not sufficiently distinctive, the overall QRS area may still differ noticeably. Therefore, introducing the QRS area into the feature space helps supplement the beat representation from a more global dynamic perspective.

After all the above extensions are introduced, we finally form a candidate feature pool composed of baseline morphological features, rhythm-related features, enhanced morphological features, and LC dynamic features. Compared with the baseline, this candidate pool is more complete in terms of information sources: it preserves the basic structural information within the current beat, supplements the temporal relations among neighboring beats, and further incorporates finer local morphological differences as well as LC-specific dynamic variation information. In total, the complete candidate pool contains 21 features, including 9 baseline features, 4 rhythm-related features, 5 enhanced morphological features, and 3 dynamic features. The complete candidate feature pool is summarized in Table 5. The purpose of constructing this candidate feature pool is not simply to improve classification performance by increasing the number of features, but rather to enrich the feature space with the missing discriminative information of the baseline while keeping the representation as compact as possible.

It should be emphasized that feature expansion does not imply that all newly introduced features should be retained unconditionally. The current set of 21 features forms a more complete candidate space that provides the basis for subsequent feature selection, but some degree of redundancy or repeated information may still exist within it. In other words, the task of this section is to complement the information missing from the baseline as comprehensively as possible, rather than to immediately determine the final optimal input set. Which features are truly necessary, which ones are strongly correlated with each other, and which subset should be retained under the compactness requirement still need to be determined through further

Table 5: Summary of the candidate feature pool.

Category	Feature	Description
Baseline morphology	A_P	Amplitude at the P point
Baseline morphology	A_Q	Amplitude at the Q point
Baseline morphology	A_R	Amplitude at the R point
Baseline morphology	A_S	Amplitude at the S point
Baseline morphology	A_T	Amplitude at the T point
Baseline morphology	Δt_{PQ}	Time interval from P to Q
Baseline morphology	Δt_{QR}	Time interval from Q to R
Baseline morphology	Δt_{RS}	Time interval from R to S
Baseline morphology	Δt_{ST}	Time interval from S to T
Rhythm	RR_{pre}	RR interval between the current and previous beats
Rhythm	RR_{post}	RR interval between the current and next beats
Rhythm	ΔRR	Difference between post-RR and pre-RR
Rhythm	RR_{mean}	Mean of the previous eight RR intervals
Enhanced morphology	D_{QR}	Amplitude difference between Q and R
Enhanced morphology	D_{RS}	Amplitude difference between R and S
Enhanced morphology	D_{ST}	Amplitude difference between S and T
Enhanced morphology	D_{PQ}	Amplitude difference between Q and P
Enhanced morphology	QRS_{dur}	Duration of the QRS complex
Dynamic	D_{cross}	Crossing density around the QRS region
Dynamic	S_{max}	Maximum local slope in the QRS region
Dynamic	A_{QRS}	QRS area

redundancy analysis and importance analysis. Therefore, the next section performs a systematic feature screening and optimization on top of this candidate pool, in order to obtain a more compact feature subset suitable for the final classifier input.

3.4 Feature Optimization and Result Analysis

The previous section expanded the baseline feature set by introducing rhythm-related features, enhanced morphological features, and LC dynamic features, thereby forming a candidate feature pool containing 21 features. Compared with the initial 9-dimensional baseline representation, this candidate pool is more complete in terms of information sources, since it simultaneously covers basic intra-beat morphological information, inter-beat rhythm relationships, finer-grained local structural differences, and the dynamic variation information specific to the LC event stream. However, the increase in the number of candidate features does not imply that all of them are equally necessary. As feature expansion proceeds, two problems naturally arise. First, some features may carry highly overlapping information, which leads to redundancy. Second, some features may provide additional descriptions, but their actual contribution to classification may be limited. If all candidate features are directly fed into the classifier without further screening, this would not only weaken the compact-representation objective of our study, but also introduce unnecessary complexity, thereby affecting the stability and generalization ability of the model. Therefore, after feature expansion, the candidate feature pool must be further optimized in a systematic manner.

In our research, feature optimization is carried out in three steps. The first step is redundancy analysis, which is used to identify features that express highly similar information. The second step is feature importance analysis, which is used to evaluate whether a feature truly contributes to the classification result. The third step is subset-size validation, which is used to determine how many features should finally be retained. In other words, the purpose of this section is not only to answer which features are useful, but also to determine which features are useful, non-redundant, and compact enough for the final classifier input.

The first step is redundancy analysis. It should be emphasized that redundancy analysis does not ask whether a feature is useful for classification. Instead, it asks whether two features are describing essentially the same phenomenon. If two features vary in a very similar way across most samples, then they are likely to carry highly overlapping information. In that case, keeping both usually provides only limited additional benefit, while increasing the dimensionality of the input. To quantify this effect, the Spearman correlation coefficient is used. For two feature vectors x and y, it is defined as

$$\rho_{x,y} = 1 - \frac{6\sum_{i=1}^{N} d_i^2}{N(N^2-1)}, \tag{33}$$

where d_i denotes the difference between the rank orders of the i-th sample in the two features, and N is the total number of samples.

The reason for using Spearman correlation instead of a standard linear correlation is that the current task is more concerned with whether two features follow the same overall trend, rather than whether they are strictly linearly related. This can be understood through a simple example. Suppose that, for a set of samples, two features assign almost the same rank order to the samples, even if their actual numerical values are very different. In such a case, the two features still convey very similar information, because they distinguish the samples in nearly the same way. Spearman correlation is well suited for detecting this type of monotonic consistency, and is therefore appropriate for redundancy analysis in the present feature pool.

The correlation analysis results show that, although the candidate feature pool does not exhibit universally high correlation across all features, several locally strong correlations can still be observed. Representative feature pairs and their Spearman coefficients are listed in Table 6. Among them, the strongest correlation is observed between the dynamic features $S_{\max}$ and D_{cross}, with a coefficient of 0.94. This result is consistent with their physical definitions. In LC representation, the local slope between adjacent events can be expressed as

$$S_i = \frac{\Delta V_{LC}}{\Delta t_i}, \tag{34}$$

where ΔV_{LC} is the quantization step and Δt_i is the time interval between adjacent crossing events. This means that the slope-based feature is fundamentally determined by inter-event timing, while D_{cross} reflects the event density in the same local region. As a result, the two features are naturally related.

Moreover, there is another important reason for removing $S_{\max}$ in the present work. Its computation requires a division operation, since the slope is obtained by dividing the amplitude increment by the inter-event interval. In offline analysis this is not a serious issue, but in low-power implementations division is generally less hardware-friendly than simple counting, comparison, or addition/subtraction. Therefore, when $S_{\max}$ already overlaps strongly with D_{cross} in terms of information content, retaining it becomes even less attractive from the perspective of low-power design. Based on these considerations, $S_{\max}$ is removed first, and the dimensionality is reduced from 21 to 20.

Feature 1	Feature 2	Spearman ρ
$S_{\max}$	D_{cross}	0.94
RR_{mean}	RR_{pre}	0.83
RR_{mean}	RR_{post}	0.81
A_{QRS}	T_{QRS}	0.74
A_R	D_{QR}	0.69
A_S	D_{RS}	0.67
A_T	D_{ST}	0.65
A_P	D_{PQ}	0.63

Table 6: Spearman correlation coefficients between representative feature pairs.

The potential redundancy relationships and the corresponding actions are further summarized in Table 7. This step shows that feature optimization is not simply a matter of minimizing dimensionality as aggressively as possible. Instead, the first priority is to remove features whose information overlaps strongly with others and whose implementation cost or retention value is relatively limited.

After redundancy analysis, feature importance analysis is further conducted to evaluate the actual contribution of each feature to the classification task. Here it is important to distinguish this step from the previous one. Redundancy analysis only answers whether two features are similar to each other; it does not

Feature 1	Feature 2	Reason	Action
S_{max}	D_{cross}	LC sampling mechanism + division cost	remove S_{max}
RR_{mean}	$RR_{\text{pre}}, RR_{\text{post}}$	rhythm correlation	evaluate later
A_{QRS}	T_{QRS}	statistical correlation	evaluate later
Amplitude features	Difference features	constructed relationship	evaluate later

Table 7: Potential redundancy relationships among candidate features.

answer whether a feature is useful for classification. A feature may be weakly correlated with others and still be unimportant, or it may be moderately correlated with another feature but still carry information on which the classifier heavily relies. Therefore, an additional analysis is required to determine the real effect of each feature on classification performance.

To this end, permutation importance is adopted. The idea is very intuitive: if a feature is truly important, then destroying its relationship with the samples should noticeably degrade the classifier performance. If a feature is not important, then disrupting it should have little effect. In practice, this is done by randomly shuffling the values of one feature across the validation or test set, while keeping all other features and all labels unchanged. In this way, the distribution of that feature remains the same, but its correspondence to the original samples is destroyed.

This can be understood through a simple example. Suppose a feature is RR_{pre}, and its values for four samples are $[0.82, 1.05, 0.76, 0.91]$. After permutation, the same values might become $[1.05, 0.76, 0.91, 0.82]$. The values themselves are still present, but they no longer belong to their original samples. If the classifier strongly depends on RR_{pre}, then this disruption will lead to a noticeable performance drop. If not, the effect will be small.

Let the original Macro-F1 of the classifier be denoted by $F1_{macro}^{base}$, and let the Macro-F1 obtained after permuting the j-th feature be denoted by $F1_{macro,j}^{perm}$. Then the importance of the j-th feature is defined as

$$I_j = F1_{macro}^{base} - F1_{macro,j}^{perm}. \tag{35}$$

A larger value of I_j indicates that the classifier performance degrades more after the corresponding feature is permuted, and hence that the feature is more important. Macro-F1 is used here instead of Accuracy because the present task is class-imbalanced, and Macro-F1 provides a fairer evaluation of performance across all classes, especially for minority and difficult classes.

The resulting feature importance ranking is shown in Table 8. The ranking shows that rhythm-related features, namely RR_{pre}, ΔRR, and RR_{post}, are among the most important features. This indicates that local rhythm variations play a central role in heartbeat classification, especially for distinguishing supraventricular ectopic beats from normal beats. Among morphological features, D_{QR}, D_{RS}, and T_{QRS} also exhibit high importance, suggesting that QRS structure remains a key factor, particularly for ventricular beat discrimination. Among the LC-specific features, D_{cross} and A_{QRS} also contribute to the classification task, indicating that dynamic information contained in the event stream can complement conventional morphology and rhythm features. By contrast, some absolute amplitude features and secondary temporal features, such as A_P, A_T, and T_{ST}, show very low importance, suggesting limited independent contribution.

Feature	Importance
RR_{pre}	0.029
ΔRR	0.027
RR_{post}	0.026
D_{QR}	0.022
D_{RS}	0.020
T_{QRS}	0.018
D_{cross}	0.016
A_{QRS}	0.013
D_{ST}	0.011
A_R	0.010
T_{QR}	0.009
RR_{mean}	0.008
T_{PQ}	0.006
A_Q	0.005
D_{PQ}	0.004
T_{RS}	0.004
A_S	0.003
A_T	0.002
T_{ST}	0.002
A_P	0.001

Table 8: Feature importance ranking based on permutation importance.

However, directly selecting the top-ranked features does not necessarily guarantee the optimal feature subset. The reason is that features may interact with each other. Some features that appear only moderately important individually may still provide stable gains when used together with other features. Conversely, some highly ranked features may offer limited marginal benefit if other retained features already capture similar information. Therefore, after obtaining the importance ranking, subset-size validation is further performed.

Let the ranked feature list be written as $F_1, F_2, \ldots, F_{20}$, and let the top-k feature subset be denoted by

$S_k = \{F_1, F_2, \ldots, F_k\}$. For each subset S_k, the classifier is retrained and evaluated. The corresponding results are shown in Table 9. These results show that performance improves as more features are included, but once the number of retained features reaches 13, the performance is already close to that of the full feature set. Further increasing the feature number yields only marginal improvement, whereas reducing the subset below 13 leads to a more noticeable drop in Macro-F1. Therefore, the 13-feature subset can be regarded as the smallest effective feature set that still preserves high classification performance.

#Features	Accuracy	Macro-F1
8	0.981	0.927
10	0.986	0.941
11	0.987	0.942
12	0.987	0.943
13	0.993	0.949
14	0.994	0.950
16	0.994	0.953
20	0.995	0.956

Table 9: Classification performance versus feature subset size.

To reduce the feature dimensionality while maintaining classification performance, low-importance features are removed step by step. The removal strategy follows three principles: removing the least important features first, retraining and re-evaluating the classifier after each removal stage, and stopping feature removal when noticeable performance degradation appears. In the first removal stage, A_P, A_T, T_{ST}, and A_S are discarded. In the second removal stage, D_{PQ}, T_{RS}, and A_Q are further removed. After these two stages, the feature dimension is reduced from 20 to 13. The performance at different reduction stages is summarized in Table 10. The results indicate that the feature dimension can be reduced from 20 to 13 with only minor performance degradation. This confirms that the candidate feature pool does contain several features whose contribution is limited or overlapped.

Feature Set	#Features	Accuracy	Macro-F1
After redundancy analysis	20	0.995	0.956
After removal stage 1	16	0.994	0.953
After removal stage 2	13	0.994	0.949

Table 10: Performance comparison during feature reduction.

Based on the redundancy analysis and feature importance evaluation presented above, the final optimized feature set is

$$\{RR_{\mathrm{pre}}, RR_{\mathrm{post}}, \Delta RR, RR_{\mathrm{mean}}, D_{QR}, D_{RS}, D_{ST}, T_{QRS}, D_{\mathrm{cross}}, A_{QRS}, T_{QR}, T_{PQ}, A_R\}. \tag{36}$$

This 13-dimensional feature set combines rhythm-related information, enhanced morphology, LC dynamic information, and a small number of complementary temporal and amplitude descriptors. Using this optimized feature set, the classifier achieves the performance shown in Table 11. Compared with the full 20-feature set, the performance degradation is minimal while the feature dimension is reduced by more than 35%.

Metric	Value
Accuracy	0.994
Macro-F1	0.949

Table 11: Final classification performance.

After determining the final optimized 13-feature subset, the classifier structure itself still needs to be further validated. It should be noted that the hidden-layer size of the MLP was not optimized separately at the initial baseline stage for the 9-dimensional feature set. Instead, after the final 13-dimensional feature subset was determined, an additional sensitivity analysis on the hidden-layer size was performed. The reason for this ordering is that, if classifier capacity were tuned before the feature-selection process was completed, it would become difficult to distinguish whether the later performance improvement came from the optimized feature subset itself or from the increased representation capacity of the classifier. Therefore, in this work, the feature-optimization procedure is first completed to obtain the final 13 input features, and the hidden-layer size is then further examined to evaluate the trade-off between classification performance and model compactness.

Specifically, while keeping the input feature set, data partitioning strategy, and the other training settings unchanged, the performance of the MLP classifier is further compared under different hidden-layer sizes. The purpose of this experiment is not merely to identify the highest classification accuracy, but to jointly consider classification performance, model size, and implementation complexity, so that a more suitable network configuration can be selected for the low-complexity objective of this work. When the hidden layer is too small, the model does not provide sufficient representation capacity to fully utilize the optimized 13-dimensional feature set. In contrast, when the hidden layer becomes too large, the performance improvement may become marginal, while the parameter count and computational cost continue to increase, thereby weakening the advantage of the method in resource-constrained scenarios.

Hidden Units	Accuracy	Macro-F1
8	0.986	0.928
12	0.989	0.939
16	0.991	0.944
20	0.992	0.946
24	0.993	0.947
28	0.993	0.948
32	0.994	0.949
40	0.994	0.949
48	0.994	0.949
56	0.994	0.949
64	0.994	0.949

Table 12: Influence of Hidden-layer size on classification performance with the final 13-feature set.

The results in Table 12 show that, as the number of hidden neurons increases, the classification performance improves within a certain range. However, once the hidden-layer size exceeds a moderate range, the performance gain becomes limited, while the model complexity continues to grow. This indicates that, for the optimized 13-feature input, an excessively large hidden layer does not provide performance improvement proportional to the additional implementation cost. Considering both classification performance and model compactness, 32 hidden neurons provide a reasonable trade-off among Accuracy, Macro-F1, and implementation cost, and are therefore adopted as the final MLP configuration in this work.

In summary, the feature optimization process in our research can be summarized as follows: first, an initial feature pool with 21 features is constructed; second, redundancy analysis is performed to remove highly correlated features; third, permutation importance is applied to evaluate feature contributions; fourth, feature subset-size validation is conducted to determine the optimal number of features; and finally, the optimized feature set containing 13 features is obtained. This result is not merely the subset with the best performance, but rather a relatively optimal representation that jointly considers expressive capability, compactness, and the requirements of low-power implementation.

3.5 Comparison with the Baseline and Existing Studies

After the LC representation, feature expansion, and feature optimization stages, we finally obtain a compact feature set containing 13 features. To further evaluate its effectiveness, the proposed method is compared from two perspectives. The first is the comparison with the baseline method developed in our study, in order to clarify the source of the performance improvement. The second is the comparison with representative existing studies, in order to position the proposed method under the constraints of compact representation and low-power-oriented design.

Table 13: Comparison between the baseline feature set, the full candidate pool, and the final optimized feature set.

Feature set	Dimensionality	Accuracy	Macro-F1
Baseline features	9	0.889	0.736
Full candidate pool	21	0.995	0.956
Final optimized features	13	0.994	0.949

First, compared with the baseline, the optimized feature set provides a clear and targeted improvement. The baseline uses only 9 basic morphological features, with an Accuracy of 0.889 and a Macro-F1 of 0.736. In contrast, the final 13-feature set improves the Accuracy to 0.994 and the Macro-F1 to 0.949. This result indicates that the main limitation of the baseline does not lie in the classifier itself, but rather in the incompleteness of the input representation. By introducing rhythm information, enhanced morphological information, and LC dynamic information, and then retaining only the most effective components through feature optimization, the classifier gains much stronger discriminative capability, especially for difficult classes such as S and F. Therefore, the improvement of the proposed method does not simply come from using more features, but from using a more informative and better organized feature representation.

Second, compared with the full candidate feature pool, the final 13-feature set provides a better balance between performance and compactness. The full candidate pool contains 21 features and is more complete in terms of information sources, but it also includes redundancy. After redundancy analysis, feature importance evaluation, and subset-size validation, the final 13 retained features preserve nearly the same level of performance while significantly reducing the input dimensionality. This is consistent with the design objective of our research. In other words, the goal here is not merely to maximize classification accuracy, but to maintain strong classification performance with as small an input representation as possible.

The internal comparison among the baseline features, the full candidate pool, and the final optimized feature set is summarized in Table 13. It can be seen that the baseline achieves the lowest performance, while the full candidate pool yields the best result at the cost of a much larger feature dimension. By contrast, the final optimized feature set retains only 13 features and still achieves an Accuracy of 0.994 and a Macro-F1 of 0.949. This suggests that the proposed optimization procedure successfully removes redundant or low-contribution features while preserving most of the discriminative information required for classification. A

Table 14: Comparison with representative studies in terms of classification target and feature dimensionality.

Author	Classification target	Number of features	Accuracy	Remarks
Fira et al. (2025)[30]	4-class	15	94.2%	MRMR-selected features.
Islam et al. (2025)[31]	5-class	65	99.58%	Compared 50, 65, 80, and 95 features.
Zhao et al. (2020)[29]	5-class	32	>98%	Event-driven ANN, 13.34 μW.
Xu et al. (2022)[14]	2-class	11	97.58%	P-QRS-T-based wearable watch-dog, 2.66 μW.
Duforest et al. (2022)[32]	2-class	32	97.97%	Slope-based event-driven features.
Proposed method	4-class	13	99.4%	LC-based compact representation.

further comparison with representative existing studies is summarized in Table 14. From this comparison, one direct advantage of the proposed method is that it achieves competitive performance with a relatively small number of input features. For example, Fira et al. used 15 features for a four-class problem and reported an Accuracy of 94.2%, whereas the proposed method uses only 13 features and reaches 99.4% Accuracy. Another feature-engineering study by Islam et al. achieved 99.58% Accuracy, but required 65 features in its best setting. By comparison, the proposed method attains a similar performance level with a much smaller feature set. This is particularly important for resource-constrained wearable systems, because a lower input dimensionality usually leads to reduced data movement, simpler classifier input layers, and lower implementation complexity.

At the same time, the comparison with low-power and event-driven studies also helps clarify the position of the proposed method. Existing event-driven works have already demonstrated the feasibility of wearable arrhythmia classification under strict power budgets. Compared with those works, the method proposed in our study does not rely on a more complex event-driven neural architecture, nor does it simply inherit conventional high-dimensional feature engineering. Instead, it combines LC-based event-driven signal representation with discriminative feature optimization, thereby constructing a compact and interpretable input space. In this sense, the proposed method serves as an intermediate solution between traditional feature-based classification and fully event-driven low-power intelligent systems.

Beyond feature dimensionality, the implementation-related complexity of the proposed method can also be further compared with representative low-power classifiers. In particular, the previously reported 13.34 μW event-driven ANN classifier [29] requires 592 multiplications per classification. In the present work, the total multiplication count per heartbeat consists of two parts, namely the multiplication cost of feature extraction and that of MLP inference:

$$N_{\text{mult,total}} = N_{\text{mult,feat}} + N_{\text{mult,MLP}}.$$

For the final classifier, the MLP takes 13 optimized features as input and uses 32 hidden neurons and 4 output neurons. Therefore, the multiplication count of the classifier itself is determined by the two fully connected layers. In the input-to-hidden layer, each of the 32 hidden neurons receives all 13 input features, resulting in

$$13 \times 32 = 416$$

multiplications. In the hidden-to-output layer, each of the 4 output neurons receives all 32 hidden activations, resulting in

$$32 \times 4 = 128$$

multiplications. Accordingly, the MLP inference stage requires

$$N_{\text{mult,MLP}} = 13 \times 32 + 32 \times 4 = 544$$

multiplications per heartbeat. In this calculation, only the weight multiplications in the fully connected layers are counted, while bias additions and the ReLU activation are not included.

The feature-extraction stage introduces only a small additional multiplication cost. First, the crossing-density feature D_{cross} is computed by normalizing the number of valid crossing events within a fixed QRS-centered window. Since the window length is constant, this normalization is implemented as one multiplication by a constant coefficient, and therefore contributes one multiplication per heartbeat. Second, instead of using the original interval-by-interval trapezoidal accumulation for A_{QRS}, a low-complexity geometric approximation is adopted. Let

$$Q = (t_Q, A_Q), \quad R = (t_R, A_R), \quad S = (t_S, A_S).$$

Then the triangle area formed by these three fiducial points can be written as

$$A_{\triangle QRS} = \frac{1}{2}\left|(t_R - t_Q)(A_S - A_Q) - (t_S - t_Q)(A_R - A_Q)\right|.$$

Using the previously defined descriptors

$$D_{QR} = A_R - A_Q, \qquad D_{RS} = A_R - A_S,$$

$$T_{QR} = t_R - t_Q, \qquad T_{RS} = t_S - t_R,$$

the above expression can be rewritten as

$$A_{\triangle QRS} = \frac{1}{2}\left(D_{QR} \cdot T_{RS} + D_{RS} \cdot T_{QR}\right).$$

Since

$$T_{RS} = T_{QRS} - T_{QR},$$

this area-related descriptor requires only two multiplications, together with simple subtraction and addition operations. The factor $\frac{1}{2}$ is treated as a constant scaling and is not counted as an additional multiplication. For the other retained features in the final 13-feature set, including RR_{pre}, RR_{post}, ΔRR, RR_{mean}, D_{QR}, D_{RS}, D_{ST}, T_{QRS}, T_{QR}, T_{PQ}, and A_R, the corresponding operations mainly involve direct readout, subtraction, counting, comparison, or averaging, and no additional multiplication is counted here. In particular, RR_{mean} is defined as the average of the previous eight RR intervals. In implementation, this can be obtained by summing the eight RR values and then performing a 3-bit right shift to divide by 8, so no additional multiplier is required.

Accordingly, the total multiplication count of the proposed final 13-feature classifier is

$$N_{\mathrm{mult,total}} = 544 + 1 + 2 = 547.$$

Therefore, compared with the 13.34 μW classifier [29], the proposed method reduces the multiplication count by

$$592 - 547 = 45,$$

which corresponds to

$$\frac{45}{592} \times 100\% \approx 7.6\%$$

fewer multiplications per heartbeat. Although the two methods are not fully identical in feature configuration and task formulation, this comparison still provides a useful implementation-oriented reference, indicating that the proposed 13-feature framework maintains competitive classification performance while slightly reducing the overall multiplication complexity relative to the 13.34 μW classifier [29].

The advantages of the proposed method in our research can therefore be summarized as follows. First, the input feature set is compact. Only 13 features are retained in the final representation, which is smaller than that of several related feature-based studies. Second, the classification performance remains competitive. On the four-class task, the proposed method achieves an Accuracy of 99.4% and a Macro-F1 of 94.9%, indicating that the reduction in input dimensionality does not lead to a substantial loss in classification performance. Third, the method is highly interpretable. From the baseline features to feature expansion, and then to redundancy analysis and importance-based selection, the introduction and retention of each feature can be clearly justified. Finally, the proposed method is more suitable for low-power implementation. Its optimization objective is not only to improve classification performance, but also to control the input dimensionality and avoid unnecessary computational complexity, which is more consistent with the practical constraints of wearable ECG classification systems.

It should also be noted that the comparison with existing studies is not strictly one-to-one. Different works may vary in data partition strategy, patient-specific or non-patient-specific settings, classification targets, and evaluation metrics, and therefore their numerical results cannot be mechanically ranked. Nevertheless, even with these differences, the position of the proposed method remains clear: it achieves competitive classification performance under the constraints of low dimensionality, interpretability, and low-power-oriented implementation. This is also consistent with the overall objective of this thesis, namely, to perform effective classification while reducing both the input scale and the subsequent computational burden as much as possible.

In summary, the LC-based compact feature representation proposed in our research not only significantly outperforms the baseline method that relies solely on basic morphological features, but also achieves competitive results while maintaining a low input dimensionality and controllable implementation complexity. This indicates that the two-stage compression strategy based on event-driven representation and discriminative feature optimization can effectively reduce the input scale of the subsequent classifier without causing a substantial sacrifice in classification performance, thereby providing a feasible path toward low-power ECG classification.

4 Event-driven ECG Classification Using Time-aware Spiking Neural Networks

Building on the feature-based LC ECG classification framework developed in Chapter 3, we further investigate whether heartbeat classification can be performed directly on LC event sequences without relying on manually designed features. The results of Chapter 3 have shown that, under low-complexity constraints, LC representation can not only reduce the redundancy of raw ECG signals, but also provide a compact and discriminative input space for subsequent classification. However, that framework still fundamentally depends on handcrafted feature engineering, in which the information ultimately presented to the classifier is determined by prior knowledge and empirical design. As a result, the richer temporal relationships and local dynamic patterns contained in the LC event stream may not be fully exploited. To address this limitation, we shift the focus from feature engineering to event-driven neural modeling. Considering the strict constraints of low-power wearable ECG monitoring, a time-aware spiking neural network is introduced to directly process asynchronous LC event streams, while incorporating inter-event timing into neuron-state evolution. In this way, the proposed framework aims to preserve the advantages of event-driven computation while making fuller use of the temporal structure inherent in LC ECG signals.

4.1 From Handcrafted Features to Event-driven Learning

The study in Chapter 3 has shown that LC-based ECG signals can be effectively classified by constructing a compact set of handcrafted features. By detecting fiducial points from the LC representation and further extracting rhythm-related, morphological, and dynamic information, it is possible to establish a low-dimensional yet highly discriminative feature space for heartbeat classification. The experimental results demonstrated that such a feature-driven framework can achieve high classification accuracy while maintaining a compact model structure, indicating that handcrafted-feature-based LC ECG classification is feasible under low-power constraints.

However, from a methodological perspective, this framework still has a fundamental limitation: it relies on a manually defined feature space. No matter how carefully the features are optimized, the information finally retained is still the result of deliberate human selection based on prior knowledge, signal understanding, and task-specific assumptions. In other words, the classifier does not learn directly from the original LC event sequence, but instead performs classification in a representation space that has already been compressed and organized by hand. This strategy has clear advantages, including structural clarity, controllable dimensionality, and relatively strong interpretability. Nevertheless, its expressive capability remains constrained by prior design, and thus cannot fully escape the limitations of empirical feature engineering. In particular, for ECG signals with complex temporal dependencies and local structural variations, predefined features may not be able to cover all information useful for classification.

This naturally raises a further question: if neural networks are capable of learning task-relevant representations directly from data [33], would it be possible to bypass handcrafted feature construction and learn discriminative patterns directly from LC ECG event sequences? Compared with feature engineering, the advantage of a data-driven model is that it does not require the designer to specify in advance which signal characteristics should be retained. Instead, the model may learn task-relevant patterns automatically during training [33, 34, 35]. Therefore, if the model can operate directly on the LC event stream, it may capture temporal dependencies and local variation patterns that are difficult to describe completely using handcrafted features, thereby making fuller use of the information carried by the LC representation itself.

This question is particularly meaningful because the representation of LC ECG signals differs fundamentally from that of uniformly sampled ECG. For uniformly sampled ECG, information is usually represented as a regularly sampled sequence of waveform points, and neural networks process dense numerical inputs defined on a fixed temporal grid. In contrast, under LC representation, the signal is converted into an asynchronously generated sequence of crossing events. Each event reflects the local direction of signal variation, while the interval between neighboring events implicitly encodes the rate of waveform change. The former describes the type of local variation, whereas the latter reflects how rapidly the signal evolves. In other words, the LC event stream inherently contains both event-type information and event-timing information. If it is further compressed into a small set of handcrafted features, part of this fine-grained temporal structure may be weakened or even lost during the compression process. Therefore, moving from handcrafted features to direct learning from event streams is not merely a change of classifier, but rather a methodological shift in how the information contained in LC signals is utilized.

However, under the low-power wearable scenario considered in this work, such a shift does not imply that any neural network model can be adopted directly. Continuous ECG monitoring systems usually operate under strict constraints in energy consumption, memory usage, and real-time processing. As a result, model design cannot focus on classification accuracy alone, but must also consider implementation cost [36, 37]. Many deep learning models, such as convolutional neural networks, recurrent neural networks, and other ar-

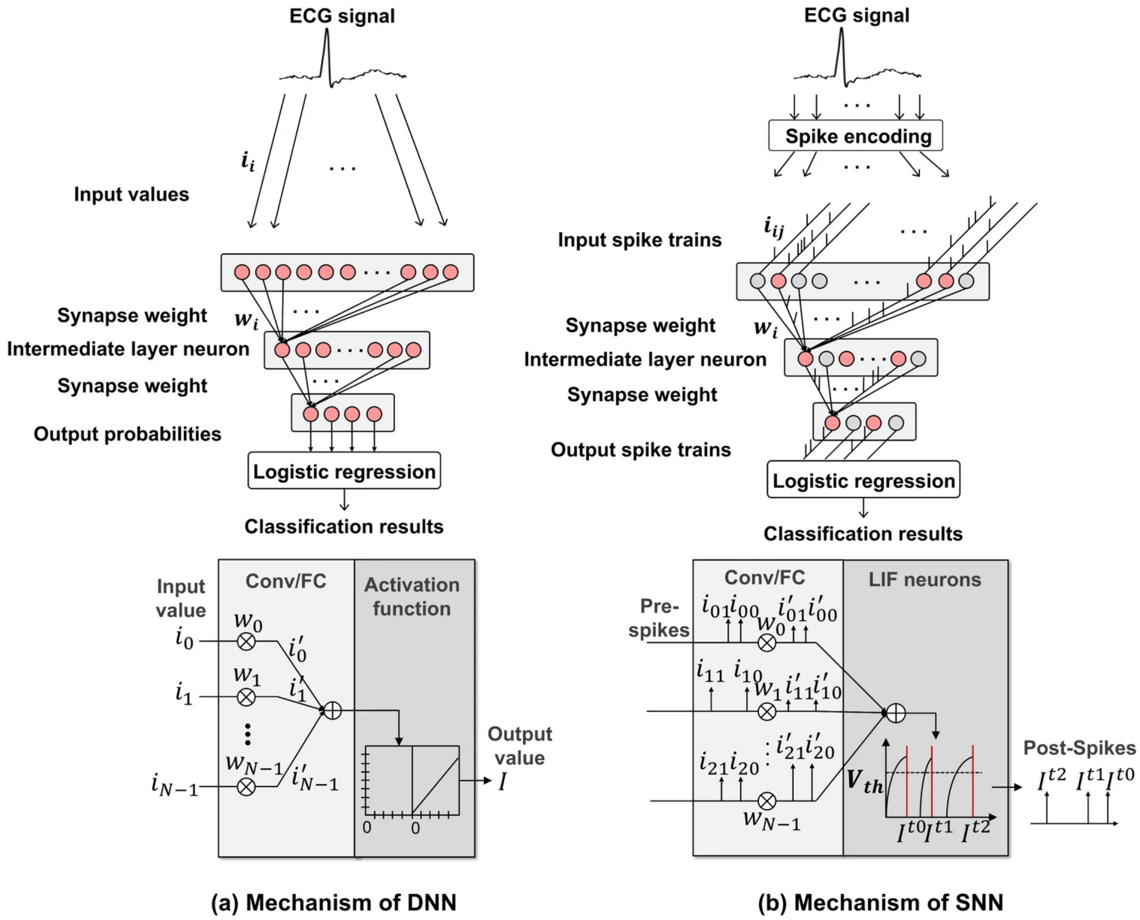

Figure 13: The Comparison between Deep Neural Network and Spiking Neural Network

chitectures based on regular time-step processing, have shown strong representation-learning ability in ECG classification [34, 35]. Nevertheless, these models are generally built upon dense numerical computation and fixed-step processing, and their computation paradigm remains more naturally aligned with uniformly sampled signals. Even after lightweight redesign or compression, they still largely preserve the dense layer-by-layer update pattern of conventional artificial neural networks, and are therefore not inherently well matched to asynchronous, sparse, and event-driven LC inputs [36, 37].

Under these constraints, spiking neural networks (SNNs) provide a more suitable alternative [38, 39]. Figure 13 shows the difference between deep neural networks(DNNs) and SNNs [11]. Unlike deep neural networks, SNNs use discrete spikes as the basic information carrier, and neuron states are typically updated only when events arrive. This gives SNNs a more event-driven computational characteristic when processing sparse inputs, which is also more compatible with the requirements of low-power implementation [38, 39]. More importantly, the membrane-potential dynamics of spiking neurons naturally involve accumulation and decay, which provide a built-in temporal mechanism for modeling irregular intervals between events [38, 11]. For LC ECG signals, the interval between neighboring crossing events directly reflects the speed of waveform variation: a short interval corresponds to a rapid local change, while a longer interval indicates a slower variation. If these temporal intervals are directly incorporated into neuron-state updates, the model can exploit not only the event type itself, but also the temporal relationships between events. This is precisely one type of information that is difficult to preserve completely in a handcrafted-feature pipeline.

Therefore, we move beyond the feature-driven framework developed in Chapter 3 and further investigate an event-driven learning approach for LC ECG classification. Instead of first compressing each heartbeat into a fixed-dimensional handcrafted feature vector and then feeding it to a classifier, the proposed method directly models the LC event sequence itself. Specifically, LC crossing events are first transformed into a discrete representation suitable for spike-based processing, and a time-aware spiking neural network classifier is then constructed on this basis. In this framework, the actual temporal interval between neighboring events is not appended as an additional handcrafted input feature, but is instead incorporated naturally into the network dynamics through the membrane decay process of spiking neurons. In this way, the model is expected to preserve the advantages of event-driven computation while learning the temporal structure and local variation patterns of LC event streams more directly.

4.2 LC Event Representation and Spike Encoding

Based on the above analysis, we propose a time-aware SNN classification framework for LC ECG event streams, whose core idea is to directly model event-driven inputs while explicitly exploiting real inter-event timing for heartbeat classification. As illustrated in Fig 14, the proposed framework mainly consists of LC event representation, local-dynamics discretization and spike encoding, time-aware SNN modeling, and output classification. The following section first introduces the input representation and spike-encoding procedure.

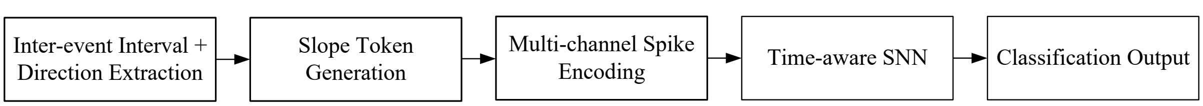

Figure 14: Overall framework of the proposed LC event-driven time-aware SNN classifier.

Before constructing the input representation, it is necessary to first clarify the basic input requirement of a spiking neural network. Unlike conventional artificial neural networks, which typically take continuous-valued vectors as inputs, SNNs use discrete spike events as the basic information carrier. Information transmission in the network is therefore mainly organized around two aspects: when an event occurs and which input channel it belongs to. A typical SNN generally consists of an input layer, a spiking hidden layer, and an output layer. The input layer receives spike-event sequences, the hidden layer processes temporal information through membrane-potential accumulation, leakage, and threshold-triggered firing, and the output layer produces the final classification result based on the network response. Precisely because SNN computation is naturally designed for discrete events rather than continuous-valued sequences, the original LC ECG event stream must first be reorganized into a form suitable for spike-based processing before being fed into the network. In other words, the purpose of this section is not yet to discuss the classifier itself, but to construct an input representation that preserves the event-driven nature of LC signals while capturing their local dynamic variations.

To enable direct SNN-based modeling of LC ECG signals, the first issue to be addressed is therefore not the classifier structure itself, but how the original LC event stream should be represented as an input suitable for spike processing. In Chapter 3, LC representation mainly served as the basis for subsequent feature extraction, where its role was to compress the original ECG waveform in an event-driven manner and to support fiducial-point detection and handcrafted-feature construction. Unlike our previous work, however, LC representation is no longer treated merely as an intermediate form before feature engineering, but is directly regarded as the raw input object for subsequent neural modeling. Accordingly, the focus of this section is no longer on extracting handcrafted features from LC signals, but on converting the LC event stream into a discrete input representation that is more suitable for spiking neural network processing while preserving its event-driven characteristics as much as possible.

Unlike uniformly sampled ECG, LC ECG is not composed of a sequence of continuous-amplitude samples acquired at fixed time intervals, but instead consists of a sequence of asynchronously generated crossing events. In other words, the basic information unit in LC representation is no longer "the signal amplitude at a fixed moment," but rather "the occurrence of a crossing event at a specific time." Therefore, in our study, the i-th LC event is represented as

$$e_i = (t_i, d_i), \tag{37}$$

where t_i denotes the time stamp of the event and d_i indicates the crossing direction. When the signal crosses from a lower quantization level to a higher one, the event is treated as an up-crossing; conversely, when the signal crosses from a higher quantization level to a lower one, it is treated as a down-crossing. The direction variable can thus be written as

$$d_i = \begin{cases} +1, & \text{up-crossing,} \\ -1, & \text{down-crossing} \end{cases}. \tag{38}$$

Accordingly, the entire LC ECG signal can be expressed as a time-ordered event sequence

$$E = \{e_1, e_2, \dots, e_N\}. \tag{39}$$

The main difference between this representation and a conventional ECG waveform lies in the fact that the signal is no longer explicitly given by regularly sampled points, but is instead implicitly reflected through the occurrence time and direction of each event. Therefore, if the LC event stream is to be used directly for classification, it becomes necessary to reconsider which information should be preserved. For a uniformly sampled waveform, local morphological information is usually expressed by the relationships among consecutive sampled values. For an LC event stream, however, local waveform variations are mainly reflected at two levels. The first is the event direction, which indicates whether the local signal is rising or falling. The second is the time interval between neighboring events, which reflects the rate of waveform variation. In other words, the information carried by LC representation does not primarily depend on a single event itself, but is more strongly embedded in the temporal relationship among events.

Based on this observation, we first consider the temporal interval between neighboring events. Let the time stamps of the i-th and $(i+1)$-th events be t_i and t_{i+1}, respectively. Then the interval between them is defined as

$$\Delta t_i = t_{i+1} - t_i. \tag{40}$$

Here Δt_i is used to describe the temporal distance between adjacent crossing events. Since LC events are triggered only when the signal crosses quantization levels, Δt_i actually reflects the rate of change of the underlying ECG waveform in the corresponding local region. When the waveform changes rapidly, for example around the QRS complex, the signal crosses multiple quantization levels within a short time, and events become denser, resulting in smaller Δt_i. In contrast, when the waveform changes slowly, such as in relatively flat baseline segments or ST–T regions, the interval between neighboring events becomes larger. Therefore, although Δt_i is not a direct amplitude measurement, it can still characterize the local variation speed of the waveform in an event-driven manner.

Besides the inter-event interval, the event direction is also indispensable. The quantity Δt_i alone only describes the variation speed, but cannot distinguish whether the signal is rising rapidly or falling rapidly. This ambiguity is compensated by the direction variable d_i. When the two are combined, a more complete description of local waveform dynamics can be formed: a small Δt_i together with a positive crossing direction indicates a rapid rising segment, whereas a small Δt_i together with a negative direction corresponds to a rapid falling segment. Larger values of Δt_i, on the other hand, usually correspond to more slowly varying regions. Therefore, by jointly considering inter-event interval and direction, local dynamic waveform behavior can be characterized without reconstructing the original uniformly sampled signal.

Furthermore, in LC quantization, the voltage difference between two adjacent crossing levels is usually fixed. Denoting this fixed level spacing by ΔV, the local slope corresponding to two adjacent crossing events can be approximated as

$$s_i = \frac{\Delta V}{\Delta t_i}. \tag{41}$$

Since ΔV is fixed under a given LC configuration, the slope magnitude is mainly determined by Δt_i. In other words, in the LC domain, a smaller inter-event interval corresponds to a larger local slope, whereas a larger interval corresponds to a smaller slope. Under this relationship, Δt_i can be regarded as a compact representation of local waveform variation strength. It should be noted that the basic idea of representing LC event streams through local dynamic variations is inspired by previous slope-based event-driven ECG classification studies. In particular, prior work has shown that local slope information derived from neighboring crossing events can effectively characterize ECG waveform variation under an event-driven framework and can be used for subsequent arrhythmia classification [32]. In the present work, however, the continuous slope is not directly used as the final handcrafted feature input. Instead, based on the same intuition, it is further combined with direction information and discretization so that the continuous local dynamics can be reorganized into a discrete token sequence more suitable for spiking neural network processing. In this way, local dynamic patterns in the LC event stream can be described directly through the combination of direction and inter-event interval, without reconstructing the continuous waveform, while also preparing the representation for subsequent spike-input construction.

Nevertheless, directly using the raw continuous values of Δt_i as network inputs would introduce two problems. On the one hand, it would make the representation highly sensitive to numerical fluctuations and noise. On the other hand, it would not be conducive to constructing a compact and stable event-driven input format. In practical ECG signals, neighboring event intervals may vary significantly due to noise, quantization jitter, and subject-specific differences. Preserving all continuous interval values would therefore not only reduce the robustness of the representation, but also weaken the simplicity of subsequent spike encoding. For this reason, we do not directly use the continuous values of Δt_i, but instead discretize them into several categories so as to obtain a more compact and robust description of local dynamics.

Specifically, three discretization thresholds are introduced to divide the range of Δt_i into four intervals. Smaller inter-event intervals correspond to faster local variations, medium intervals correspond to moderate variation speeds, and larger intervals correspond to slower or nearly flat local segments. The purpose of this discretization is not to achieve precise numerical fitting of the slope magnitude, but rather to distinguish different types of local dynamic patterns using a limited number of discrete categories. In this way, the representation becomes less sensitive to fine-grained numerical perturbations, while the overall input-space complexity is also controlled. As a result, the subsequent network can focus more on category-level differences in local dynamic patterns instead of relying excessively on exact numerical values.

After discretizing the inter-event intervals, we further combine the resulting interval categories with direction information to construct discrete slope tokens. Specifically, local waveform variations are encoded into the following seven symbols:

$$\{+++,++,+,N,-,--,---\}. \tag{42}$$

Here, $+++$, $++$, and $+$ represent different degrees of rising variations; $-$, $--$, and $---$ represent different degrees of falling variations; and N denotes a nearly flat local segment. Through this representation, the continuously varying local dynamic patterns in the original LC event stream are reorganized into a sequence of discrete symbolic tokens. The resulting token sequence no longer depends directly on the original waveform amplitudes, but instead provides a more direct description of the type, direction, and relative speed of local variations.

From the perspective of representation design, the introduction of slope tokens has two important implications. First, it transforms the original continuous and potentially sensitive inter-event interval information into a limited number of stable categories, thereby improving representation robustness. Second, it provides a natural interface for constructing inputs to the spiking neural network. Since SNNs are better suited to discrete events than to continuous-valued streams, encoding LC local dynamics into discrete tokens helps preserve the event-driven nature of the original signal while aligning the input form with the subsequent network-processing mechanism. In this sense, the token sequence here is not a handcrafted feature vector, but rather an intermediate representation tailored for spike-based computation, whose role is to convert the LC event stream into a discrete temporal form more suitable for SNN modeling.

Once the slope-token sequence has been obtained, the next step is to map it into spike inputs. To this end, a multi-channel spike-encoding scheme is adopted, in which each token is mapped to an independent input channel. Let the input spike vector be

$$\mathbf{x}_i = [x_i^1, x_i^2, \ldots, x_i^C], \tag{43}$$

where C denotes the number of input channels. Since seven token categories are defined in our work, the number of input channels is set to 7. When the i-th token occurs, only the input channel corresponding to that token is activated, while all other channels remain silent. That is,

$$x_i^k = \begin{cases} 1, & k = \mathrm{channel}(T_i), \\ 0, & \text{otherwise} \end{cases}. \tag{44}$$

As a result, each local dynamic token is represented as a one-hot spike event, and the entire LC ECG signal is further converted into a multi-channel spike sequence ordered by event occurrence:

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}. \tag{45}$$

It should be emphasized that, during this process, the original asynchronous event stream is not converted into a uniformly sampled spike train with fixed time steps, nor is the temporal interval information discarded. Instead, the event time stamps are still preserved, while the multi-channel spike vector is only used to represent what type of local variation has occurred. The questions of when these variations occur and how far apart neighboring variations are remain represented by the original event-time information. In other words, in the input representation constructed in our research, the token channels encode local variation types, whereas the event time stamps preserve the asynchronous temporal structure. Together, they constitute the input basis required by the subsequent time-aware SNN.

Through the above procedure, the original LC ECG signal is ultimately reorganized into an asynchronous input representation suitable for spiking neural network processing. On the one hand, local waveform variations are converted into multi-channel spike inputs through discrete slope tokens. On the other hand, the actual temporal intervals between events are still preserved for subsequent neuron-dynamics modeling. The resulting representation not only maintains the essential characteristics of LC signals, namely event-driven, asynchronous, and sparse processing, but also avoids the instability that would arise from directly using continuous inter-event values as raw inputs. Moreover, building upon the basic idea of slope-based event representation, it further restructures the LC event stream into a form tailored for SNN input, thereby laying the foundation for the construction of the time-aware spiking neural network model in the next section.

4.3 Time-Aware SNN Classifier Construction

In the previous section, the original LC ECG signal was reorganized into an asynchronous multi-channel spike representation, where local waveform variation types were mapped to spike-input channels through discrete slope tokens, while the actual temporal intervals between neighboring events were preserved as an essential basis for subsequent temporal modeling. On this basis, this section further constructs a classifier that can directly process such inputs. It should be emphasized that, if the subsequent network still adopts a conventional fixed-time-step processing scheme, then even though the input has already been represented as an event sequence, the most critical irregular temporal structure in the LC event stream may still be weakened under clock-driven updates. Therefore, instead of employing a conventional fixed-step spiking neural network, we introduce a time-aware SNN model in which the actual temporal interval between neighboring events directly participates in neuron-state evolution.

From the perspective of the modeling objective, the LC ECG event stream contains two complementary types of information. The first is the event type itself, namely, which local dynamic token the current event belongs to. The second is the temporal relationship between events, namely, how long after the previous local variation the current one occurs. The former has already been encoded in Section 4.2 through multi-channel spike inputs, whereas the latter should not be simply concatenated as an ordinary auxiliary feature. The reason is that the temporal interval is not merely a static attribute parallel to event type; rather, it determines how strongly neighboring events interact over time. If a new event arrives shortly after the previous one, the influence of the previous event on the current neuron state should be preserved to a greater extent. In contrast, if two events are separated by a longer interval, the previous influence should undergo more substantial decay. Therefore, the temporal interval is better incorporated directly into neuron dynamics than appended as a static feature to a downstream classifier. This is also the central motivation for introducing a time-aware SNN in our research.

To achieve this goal, the leaky integrate-and-fire (LIF) neuron is adopted as the basic computational unit. The LIF model is one of the most widely used neuron models in spiking neural networks. Its key idea is to use the membrane potential to represent the accumulated response of a neuron to incoming events: when a new input arrives, the neuron integrates the input; during periods with no new input, the membrane potential gradually decays toward the resting state; once the membrane potential rises above a predefined threshold, the neuron emits a spike and its state is reset. Because the LIF model naturally includes the process of accumulation, leakage, and triggering, it is particularly suitable for processing asynchronous event streams with explicit temporal structure.

In continuous time, the membrane-potential dynamics of an LIF neuron can be written as

$$\tau_m \frac{dV(t)}{dt} = -\left(V(t) - V_{rest}\right) + RI(t), \tag{46}$$

where $V(t)$ denotes the membrane potential at time t, V_{rest} is the resting potential, $I(t)$ is the input current, R is the membrane resistance, and τ_m is the membrane time constant. For the LC ECG event stream considered in our study, however, inputs do not arrive continuously under regular time steps, but occur asynchronously as discrete events. Therefore, if Eq. (46) were still updated using a fixed global clock, this would not only introduce unnecessary idle updates, but also fail to fully exploit the effect of real inter-event timing on neuron states. To address this issue, this work adopts an event-driven state-update mechanism, in which neuron states are updated only when a new event arrives, while between two neighboring events the membrane potential undergoes only passive decay.

Let the arrival time of the i-th input event be t_i, and that of the previous event be t_{i-1}. The corresponding inter-event interval is denoted by $\Delta t_i = t_i - t_{i-1}$. During the interval between two adjacent events, when no new external input is considered, the membrane potential evolves only according to passive leakage, whose analytical solution can be written as

$$V(t) = V_{rest} + (V(t_0) - V_{rest})\, e^{-\frac{t-t_0}{\tau_m}}. \tag{47}$$

Accordingly, when the i-th event arrives, the membrane potential can be directly written in the following event-driven update form:

$$V_i = V_{rest} + (V_{i-1} - V_{rest})\, e^{-\frac{\Delta t_i}{\tau_m}} + \mathbf{W}\mathbf{x}_i, \tag{48}$$

where $\mathbf{x}_i$ denotes the i-th input spike vector and $\mathbf{W}$ is the corresponding synaptic weight vector. If the resting potential is normalized to zero, namely $V_{rest} = 0$, Eq. (48) can be simplified as

$$V_i = V_{i-1} e^{-\frac{\Delta t_i}{\tau_m}} + \mathbf{W}\mathbf{x}_i. \tag{49}$$

Equation (49) is the core of the time-aware mechanism proposed in our research. Unlike a fixed-step SNN, where the amount of decay is determined by a uniform simulation step, here the membrane decay is directly governed by the actual inter-event interval Δt_i. In other words, when two neighboring events are close in time, the exponential decay is weaker and the effect of the previous event is retained to a greater extent. When two events are separated by a longer interval, the membrane potential decays more strongly, and the contribution of the earlier input is naturally weakened. In this way, the asynchronous temporal structure of the LC event stream is directly embedded into neuron dynamics without constructing any additional inter-event timing feature vector.

After the updated membrane potential is obtained, spike generation is determined by a thresholding rule:

$$S_i = \begin{cases} 1, & V_i \geq V_{th}, \\ 0, & V_i < V_{th} \end{cases}, \tag{50}$$

where V_{th} is the firing threshold. When the membrane potential exceeds the threshold, the neuron emits an output spike. After firing, to prevent unlimited accumulation of the membrane potential and distortion of subsequent dynamics, a hard reset strategy is adopted to return the neuron state to a predefined reset potential

V_{reset}:

$$V_i = \begin{cases} V_{reset}, & S_i = 1, \\ V_i, & S_i = 0 \end{cases}. \tag{51}$$

In our study, V_{reset} is typically set to 0. Together, Eqs. (49), (50), and (51) constitute the complete update mechanism of the time-aware LIF neuron.

Based on the above neuron model, a compact time-aware SNN classifier is further constructed for heartbeat classification. The overall network adopts a three-layer architecture consisting of an input layer, a hidden spiking layer, and an output layer. The input layer receives the multi-channel spike sequence defined in Section 4.2, in which each event is represented by a one-hot vector encoding its local dynamic type while its event time is preserved. The hidden layer is composed of time-aware LIF neurons, whose task is to integrate the input-event sequence over time. Since the hidden-neuron update explicitly depends on Δt_i, the network is able to jointly model both event-type information and inter-event timing information in the LC ECG stream. The output layer then performs the final classification based on the hidden-layer responses.

Let the number of input channels be C, the number of hidden neurons be H, and the number of output classes be K. For the j-th hidden neuron, its membrane-potential update upon the arrival of the i-th event can be written as

$$V_i^{(j)} = V_{i-1}^{(j)} e^{-\frac{\Delta t_i}{\tau_m}} + \sum_{k=1}^{C} w_{jk} x_i^k, \tag{52}$$

where w_{jk} denotes the synaptic weight from input channel k to hidden neuron j, and x_i^k denotes the spike state of the i-th input event on channel k. This equation shows that each hidden neuron evolves independently in an event-driven manner, while all neurons share the same inter-event timing information and respond to the temporal structure of the event stream.

In the output layer, we adopt a spike-count-based readout, meaning that output neurons accumulate spikes generated over the entire input-event sequence rather than relying on a single instantaneous response. Let the accumulated response of the c-th output neuron over the whole sequence be denoted by z_c. Then

$$z_c = \sum_{i=1}^{M} S_i^{(c)}, \tag{53}$$

where $S_i^{(c)}$ indicates whether the c-th output neuron fires at the i-th event. The final four-class prediction is determined according to the accumulated responses of all output neurons. The advantage of this readout strategy is that it aggregates the dynamic response over the entire event sequence rather than depending on a single local state, which is more suitable for asynchronous event-sequence classification.

In addition to the modeling capability, network compactness is also a major design consideration in our design. Since the focus of this work is low-power wearable ECG monitoring, the model is not designed through unconstrained depth expansion or parameter scaling. Instead, it aims to maintain strong classification performance while controlling parameter size and computational complexity. If the number of input channels is 7, the number of hidden neurons is H, and the number of output classes is 4, then the network contains $7H$ input-to-hidden weights and $4H$ hidden-to-output weights. If the hidden-layer and output-layer biases are further included, the total number of parameters can be written as

$$N_{param} = 7H + 4H + H + 4, \tag{54}$$

where $7H$ corresponds to the number of input-to-hidden weights, $4H$ corresponds to the number of hidden-to-output weights, and H and 4 correspond to the hidden-layer and output-layer bias parameters, respectively. Compared with conventional deep networks based on convolutions or long-sequence unfolding, this structure is considerably more compact and is therefore more consistent with the low-complexity design objective of this work.

In summary, based on the asynchronous spike-input representation constructed in Section 4.2, this section further proposes a time-aware SNN classifier. Its core contribution does not lie in simply feeding LC event streams into a spiking network, but in allowing the actual temporal intervals between neighboring events to directly influence neuron-state evolution through event-driven membrane updates. In this way, the two most representative types of information in LC ECG signals, namely local variation type and asynchronous temporal structure, can be jointly expressed within a single neural model. This provides the complete model basis for the subsequent training strategy and experimental validation.

4.4 Training and Experimental Setup

In the previous section, a time-aware SNN classifier for LC ECG event streams was constructed. To further evaluate its effectiveness on heartbeat classification, this section provides a unified description of the training strategy, dataset and task setting, evaluation metrics, and implementation details. Since the proposed model is based on asynchronous event sequences and spiking neuron dynamics, its training procedure differs substantially from that of conventional feedforward neural networks. It is therefore necessary to first clarify how the model performs forward propagation and parameter optimization under an event-driven framework, before introducing the data source, task definition, and hyperparameter configuration used in the experiments.

For each input heartbeat, the model does not receive a fixed-length continuous-valued sequence, but rather an asynchronous event sequence generated by the encoding procedure in Section 4.2,

$$\{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots, (\mathbf{x}_M, t_M)\},$$

where $\mathbf{x}_i$ denotes the multi-channel spike vector of the i-th input event, t_i is its occurrence time, and M is the total number of events contained in the heartbeat. It should be noted that different heartbeats produce different numbers of LC events, so each input sample is essentially a variable-length event sequence. During forward propagation, the network is unfolded according to the order of event arrivals: whenever a new event arrives, the membrane potential of each hidden neuron is first decayed according to the actual temporal interval between the current and previous events, then updated by the synaptic response induced by the current input, and finally checked against the firing threshold to determine whether an output spike should be generated. The output layer accumulates spike responses over the whole event sequence and produces the final classification result accordingly. In other words, forward propagation in the proposed model is not performed by scanning along a uniformly discretized simulation timeline, but by updating the network state only when real events occur. Its effective unfolding length is therefore determined by the number of input events rather than by a predefined time step.

Since different samples have different event lengths, the training implementation must explicitly handle length variation within each mini-batch. In this work, event sequences are aligned to a common batch length, and a masking mechanism is used so that only valid events contribute to state updates, while padded positions are ignored. This design preserves the variable-length nature of event-driven inputs while still enabling efficient parallel training.

However, unlike conventional artificial neural networks, the spike-generation process in an SNN involves a non-differentiable step function, which prevents the direct application of standard backpropagation. To address this issue, surrogate gradients are adopted to approximate the gradient of the spike function, thereby enabling end-to-end training through gradient descent. Let the spike state of a neuron be defined by a thresholding function,

$$S_i = H(V_i - V_{th}),$$

where $H(\cdot)$ denotes the Heaviside step function. Since this function is not differentiable at the threshold, a smooth surrogate is used during backpropagation. Specifically, the derivative is approximated as

$$\frac{\partial S_i}{\partial V_i} \approx \frac{1}{(1 + \beta |V_i - V_{th}|)^2}, \tag{55}$$

where β is a hyperparameter controlling the slope of the surrogate gradient. The role of this approximation is to preserve discrete spiking behavior during forward propagation while providing nonzero gradients around the threshold during backward propagation, so that gradients can pass through the spike-generation process and update the network parameters.

Because the membrane-potential state of the time-aware LIF neuron depends on historical inputs as well as their inter-event timing, training must still unfold the network along the temporal dimension. In this work, Backpropagation Through Time (BPTT) is adopted for optimization. The basic idea of BPTT is to unfold a temporally dependent network into a longer computational graph along the time axis and then perform gradient backpropagation over the unfolded graph, so that the current loss can update shared parameters through their effects on historical states. In the present model, the membrane potential at the current event depends not only on the current input, but also on the state-evolution chain formed by preceding events. Therefore, BPTT is required to correctly propagate the error signal back to earlier event updates. It should be emphasized that, unlike conventional fixed-step SNNs, the unfolding here is not based on a uniform clock-driven grid, but on the input event sequence itself. In other words, the network is unfolded along event indices rather than artificial simulation steps. This design allows the model to directly exploit the dynamic effect induced by real inter-event intervals during training, while avoiding the large number of idle updates that would arise in a fixed-step framework.

In the loss-function design, the objective is not only to optimize classification accuracy, but also to regulate neuron activity and membrane-potential stability. The overall loss function is defined as

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_s \mathcal{L}_{spike} + \lambda_v \mathcal{L}_{volt}, \tag{56}$$

where $\mathcal{L}_{cls}$ denotes the classification loss, $\mathcal{L}_{spike}$ is the spike-activity regularization term, and $\mathcal{L}_{volt}$ is the membrane-potential regularization term. The coefficients λ_s and λ_v control the importance of the latter two terms. After the output layer accumulates spike responses over the whole event sequence, the resulting class scores z_c are passed through a softmax function to obtain class probabilities, which are then used to compute the classification loss. The classification loss is implemented as a weighted cross-entropy loss,

$$\mathcal{L}_{cls} = -\sum_{c=1}^{K} w_c y_c \log p_c, \tag{57}$$

where K is the number of output classes, y_c is the ground-truth label, p_c is the predicted probability for class c, and w_c is the corresponding class weight. In this work, the class weights are determined according to the number of training samples in each category, so that minority classes receive larger loss weights. This helps alleviate the class-imbalance problem commonly encountered in ECG datasets and reduces the tendency of the model to overfit majority classes.

To control the activity level of hidden neurons, a spike regularization term is further introduced:

$$\mathcal{L}_{spike} = \frac{1}{MH} \sum_{i=1}^{M} \sum_{j=1}^{H} S_i^{(j)}, \tag{58}$$

where M is the number of input events, H is the number of hidden neurons, and $S_i^{(j)}$ denotes the spike state of the j-th hidden neuron at the i-th event. This term penalizes excessively frequent firing, thereby encouraging sparse neural activity, which is consistent with the low-power objective of event-driven computation. In addition, to avoid excessive supra-threshold accumulation of membrane potentials during training, a membrane-potential regularization term is also included:

$$\mathcal{L}_{volt} = \frac{1}{MH} \sum_{i=1}^{M} \sum_{j=1}^{H} \max\left(0, V_i^{(j)} - V_{th}\right)^2. \tag{59}$$

This term suppresses membrane potentials from drifting too far above the threshold and helps maintain stable neuron dynamics during optimization.

The experiments in this chapter are conducted on the MIT-BIH Arrhythmia Database. To ensure strict comparability with Chapter 3, the same record selection, beat segmentation, preprocessing pipeline, data partitioning protocol, and LC encoding parameter settings are adopted here and are therefore not repeated in detail. Following the same AAMI-based grouping protocol as in Chapter 3, four heartbeat classes are considered, namely N, S, V, and F. Therefore, compared with the previous chapter, the main differences in this chapter lie not in the dataset configuration itself, but in the downstream classifier design, input modeling strategy, and the corresponding event-driven training mechanism.

With respect to evaluation metrics, Accuracy and Macro-F1 are adopted as the main classification-performance measures. Accuracy reflects the overall correctness of prediction, while Macro-F1 provides a more balanced evaluation across categories and is therefore more representative in imbalanced classification tasks. In addition, because the focus of this chapter is low-power event-driven classification, the average neuron firing rate and the approximate number of synaptic operations (SynOps) are also evaluated to characterize network sparsity and implementation complexity. Here, the average firing rate reflects the degree of spiking activity in the hidden layer, while SynOps is used as an approximate indicator of the computation cost required for one heartbeat inference. For a single heartbeat in this work, the approximate number of synaptic operations is defined as

$$SynOps \approx M \times H, \tag{60}$$

where M is the number of input events for that heartbeat and H is the number of hidden neurons. This approximation directly reflects the fact that each arriving input event triggers one state update across the hidden layer, and can therefore serve as a practical estimate of the computational scale of the event-driven model. By jointly considering classification performance and activity-related complexity, the experiments can more comprehensively assess whether the model balances accuracy with low-power implementation potential.

In terms of implementation details, the model is trained using the PyTorch framework together with custom event-driven neuron modules for time-aware state updates. The Adam optimizer is adopted with an initial learning rate of 10^{-3}, and the batch size is set to 32. Early stopping based on validation Macro-F1 is used to prevent overfitting. For the regularization coefficients in the loss function, the default values are set as $\lambda_s = 10^{-4}$ and $\lambda_v = 10^{-3}$. In the surrogate-gradient function, the slope parameter β is set to 10 to ensure stable gradient approximation during training. The model typically converges within about 100 epochs, although the exact convergence speed depends on the hidden-layer size, membrane time constant, and threshold configuration.

To determine an appropriate model configuration, several key hyperparameters are further explored, including the number of hidden neurons H, the membrane time constant τ_m, and the firing threshold V_{th}. The hidden-layer size is tested over the candidate set

$$\{8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 56, 64, 80, 96\},$$

so as to analyze the trade-off between model capacity and computational complexity. The membrane time constant is varied over

$$\{2, 4, 8, 16, 32\},$$

to evaluate how much temporal context should be retained by the neuron dynamics. The firing threshold is also examined within the range

$$\{0.5, 1.0, 1.5, 2.0\},$$

in order to compare different levels of activity sparsity and classification behavior. It should be emphasized that this section only introduces the search ranges and training settings of these parameters. Their influence on classification performance and firing activity, as well as the rationale for the final parameter selection, will be presented in detail in the next section.

On the basis of the above training strategy, dataset setting, and implementation details, the next section further presents the experimental results of the proposed time-aware SNN and analyzes its performance under different parameter configurations and baseline comparisons.

4.5 Experimental Results and Discussion

Based on the training strategy, dataset setting, evaluation metrics, and hyperparameter search ranges introduced in the previous section, this section further analyzes the influence of different parameter configurations on model performance and neural activity, and compares the proposed method with several baseline models. Since the objective of this work is not limited to classification accuracy alone, the following analysis jointly considers predictive performance, firing sparsity, and implementation complexity.

We first examine the influence of the hidden-layer size. The number of hidden neurons H determines the representational capacity of the model and also directly affects the number of state updates and the level of spiking activity. The results under different hidden-layer sizes are summarized in Table 15. When the hidden layer is relatively small, the representational capacity of the model is limited, and the classification performance is correspondingly lower. For example, when $H = 8$, the model achieves an Accuracy of 97.42%

and a Macro-F1 of 0.871. As the number of hidden neurons increases, the model becomes more capable of integrating temporal patterns from the input event stream, and the classification performance improves steadily. However, once H exceeds 32, the performance gain becomes marginal, while the average firing rate continues to increase. This indicates that a larger hidden layer introduces higher activity and computational cost without providing a proportional improvement in classification performance. Considering both predictive performance and activity-related complexity, $H = 48$ is finally selected as the default hidden-layer size, since it provides a favorable trade-off between classification accuracy and spiking sparsity.

Table 15: Influence of hidden-layer size on classification performance and firing activity.

Hidden neurons	Accuracy (%)	Macro-F1	Firing rate (%)
8	97.42	0.871	7.8
12	97.86	0.889	9.4
16	98.21	0.901	11.3
20	98.53	0.915	12.6
24	98.74	0.919	13.7
28	98.85	0.924	14.8
32	98.96	0.931	16.2
36	98.98	0.933	17.0
40	99.00	0.935	17.6
44	99.01	0.936	18.1
48	99.02	0.936	18.4
56	99.02	0.936	19.7
64	99.03	0.937	21.1
80	99.03	0.937	23.9
96	99.04	0.938	26.8

After determining the hidden-layer size, we further analyze the influence of the membrane time constant τ_m. This parameter controls how long past inputs can affect the current neuron state and therefore determines the effective temporal memory of the model. The results are shown in Table 16. When τ_m is too small, the membrane potential decays too rapidly, and the neuron cannot effectively integrate neighboring events over time. This leads to weaker classification performance. As τ_m increases from 2 to 8, the model is able to make better use of temporal dependencies in the local event sequence, and both Accuracy and Macro-F1 improve. However, when τ_m is further increased to 16 or 32, the performance begins to decline. This suggests that overly long temporal retention weakens the sensitivity of the model to local rhythm variations and rapid dynamic changes. According to the results in Table 16, $\tau_m = 8$ provides the best overall performance and is therefore adopted in the final model.

The firing threshold V_{th} is another important factor affecting both network activity and classification

Table 16: Influence of membrane time constant τ_m on classification performance and firing activity.

τ_m	Accuracy (%)	Macro-F1	Firing rate (%)
2	98.18	0.897	20.1
4	98.63	0.918	19.2
8	99.02	0.936	18.4
16	98.94	0.930	17.1
32	98.71	0.923	16.2

behavior. Table 17 presents the experimental results under different threshold settings. When the threshold is low, neurons fire more easily and the network exhibits a higher firing rate. For instance, at $V_{th} = 0.5$, the model achieves relatively strong classification performance, but the firing activity rises substantially. As the threshold increases, the spiking activity becomes sparser. However, when the threshold is set too high, firing becomes overly suppressed and the expressive capability of the model is weakened, which leads to performance degradation. The results indicate that $V_{th} = 1.0$ provides the most balanced trade-off between Accuracy, Macro-F1, and firing rate. Consequently, the final model configuration is set to $H = 48$, $\tau_m = 8$, and $V_{th} = 1.0$.

Table 17: Influence of firing threshold V_{th} on classification performance and firing activity.

V_{th}	Accuracy (%)	Macro-F1	Firing rate (%)
0.5	98.96	0.931	24.8
1.0	99.02	0.936	18.4
1.5	98.90	0.929	15.3
2.0	98.67	0.921	14.3

After fixing the model configuration, we compare the proposed method with several baseline models. The comparison results are summarized in Table 18, including the proposed time-aware SNN, a fixed-step SNN, the Feature+MLP framework proposed in Chapter 3, and a 1D CNN baseline. Among these models, the 1D CNN achieves the highest overall classification performance, with an Accuracy of 99.60% and a Macro-F1 of 0.953. This is expected, since convolutional models are highly effective at extracting local features from dense waveform inputs, although they generally require much larger model size and higher computational cost. The Feature+MLP method proposed in Chapter 3 also achieves very strong results, with an Accuracy of 99.40% and a Macro-F1 of 0.949, confirming the effectiveness of handcrafted-feature-based LC ECG classification.

Compared with these methods, the proposed time-aware SNN achieves a slightly lower overall Accuracy of 99.02% and a Macro-F1 of 0.936, but it maintains a very compact parameter scale and an event-driven computation pattern. More importantly, compared with the fixed-step SNN, the proposed time-aware SNN performs better in both Accuracy and Macro-F1, indicating that explicitly incorporating real inter-event timing into neuron-state evolution indeed improves the modeling of LC ECG temporal structure. This suggests that the proposed time-aware mechanism is not simply an architectural variation, but a meaningful improvement in the way event-driven classification models exploit asynchronous timing information.

Table 18: Comparison between the proposed method and baseline models.

Model	Accuracy (%)	Macro-F1	Parameters
1D CNN	99.60	0.953	$> 10^4$
Feature + MLP (Chapter 3)	99.40	0.949	544
Fixed-step SNN	98.70	0.920	528
Proposed time-aware SNN	99.02	0.936	528

Beyond the fair comparison under the unified experimental setting, the proposed method is further positioned with respect to recent related studies, as summarized in Table 19. It should be noted that the studies listed in the table do not follow exactly the same data split, class definition, input representation, or evaluation protocol. Nevertheless, the reported accuracies still provide a useful reference for positioning the proposed method with respect to recent related work. From this perspective, the proposed method achieves a highly competitive classification performance.

From Table 19, it can be seen that the proposed method achieves strong classification performance among recent related studies. In particular, the reported Accuracy of the proposed method reaches 99.02%, which is higher than that of most event-driven, SNN-based, or low-power ECG classification studies listed in the table. This observation indicates that the proposed framework is not only meaningful from the perspective of event-driven modeling, but is also highly effective in terms of classification performance.

More importantly, the advantage of the proposed method is not limited to a higher reported Accuracy. Compared with many existing studies, the present work does not build a lightweight classifier on top of a conventional uniformly sampled ECG waveform. Instead, it directly constructs a time-aware SNN classifier on the basis of the LC event stream itself. As a result, the proposed framework preserves the sparsity and low-redundancy characteristics of event-driven ECG representation at the input level, while also explicitly incorporating real inter-event timing into neuron-state evolution. In this sense, the proposed method does not merely replace a conventional classifier with an SNN, but establishes a more coherent event-driven pipeline that connects input representation, temporal modeling, and low-power implementation objectives within a unified framework.

Table 19: Comparison with recent related studies.

Work	Year	Input representation	Model type	Deployment orientation	Reported Accuracy
Zhao et al. [29]	2020	Event-driven ECG input	Patient-specific ANN	Wearable / ultra-low-power classification	$> 98\%$
Duforest et al. [32]	2021	Event-driven slope LC features	Feature extraction + classifier	Low-complexity / event-driven ECG classification	98.4%
Saeed et al. [26]	2021	LC-ADC event-driven ECG representation	1D CNN	Event-driven front-end feasibility study	99.49%
Xing et al. [40]	2022	Spike-encoded ECG	SNN + channel-wise attention	Portable / low-power ECG classification	98.26%
Chu et al. [41]	2022	LC-sampled event-driven ECG	Spike-driven SNN processor	Neuromorphic wearable ECG system	98.22%
Jiang et al. [42]	2024	Raw ECG waveform	Lightweight Transformer	Edge devices / limited power and storage	99.84%
Rahman et al. [43]	2025	Beat-level ECG input	Resource-constrained on-chip classifier	Real-time on-chip ECG classification	98.1%
This work	–	LC event stream + slope-token spike encoding	Time-aware event-driven SNN	Low-power wearable / event-driven framework	99.02%

Therefore, the significance of the proposed method lies in two aspects. On the one hand, it achieves strong classification performance and remains competitive, or even superior, to most related studies in terms of reported Accuracy. On the other hand, it realizes this performance while directly operating on LC event streams and maintaining an event-driven computational form. This means that the proposed method does not trade classification accuracy for methodological novelty or low-power compatibility. Instead, it demonstrates that a direct event-driven learning framework can preserve the implementation advantages of LC ECG representation while still reaching a high level of classification effectiveness.

Table 20: Per-class F1-score comparison between feature-based and the proposed method.

Class	Feature + MLP (Chapter 3)	Proposed time-aware SNN
N	0.992	0.987
S	0.938	0.910
V	0.978	0.953
F	0.888	0.894
Macro-F1	0.949	0.936

To further examine the class-wise behavior of the proposed model, Table 20 reports the per-class F1-scores of the Feature+MLP method in Chapter 3 and the proposed time-aware SNN. It can be seen that the handcrafted-feature-based method achieves higher F1-scores on the N, S, and V classes. This result is understandable, since the feature-based framework explicitly incorporates rhythm, morphology, and LC dynamic features, all of which provide strong prior guidance for separating normal beats, supraventricular ectopic beats, and ventricular ectopic beats. In contrast, the proposed method no longer relies on manually designed discriminative descriptors, and instead learns directly from the LC event stream. As a result, some class boundaries must be formed entirely through data-driven temporal modeling.

Nevertheless, the proposed time-aware SNN achieves a slightly better F1-score on the F class, improving from 0.888 to 0.894. Although the improvement is modest, it is meaningful. Fusion beats often exhibit ambiguous morphology and weakly defined boundaries, making them more difficult to characterize using explicit handcrafted descriptors. By directly modeling event-stream dynamics and inter-event timing, the proposed method is able to exploit finer-grained asynchronous local patterns, which may explain its slight advantage on this category. This observation also suggests that the value of the proposed method does not lie solely in improving overall metrics, but in exploring a more direct event-driven modeling framework that may offer specific advantages on morphologically ambiguous classes.

The confusion-matrix analysis further supports the above observations. Normal beats remain the easiest category to recognize, which is consistent with their dominant proportion in the dataset and their relatively stable morphology. Confusion between the S and N classes is still observed, which is common in ECG classification because some supraventricular ectopic beats are locally similar to normal beats and differ more in rhythm context than in isolated local morphology. Ventricular beats are recognized relatively well, mainly because they often exhibit more distinct local dynamic abnormalities around the QRS region, which can be effectively reflected in the event stream. Although the F class contains fewer samples and tends to be morphologically ambiguous, the proposed method shows a slight advantage in this category, consistent with the per-class F1 analysis. Overall, the confusion analysis indicates that the proposed time-aware SNN is already capable of capturing the major class differences in LC event streams, although its performance remains slightly below that of handcrafted-feature-based methods on categories that benefit strongly from explicit prior descriptors.

Besides classification performance, model complexity and low-power implementation potential are also major concerns in this work. For an event-driven SNN, the inference complexity is mainly determined by the number of neuron updates and synaptic operations. According to the complexity definition given in Section 4.4, under the LC configuration adopted in this work, a heartbeat typically generates about 80 to 100 events, and the hidden-layer size is fixed at 48. Therefore, the hidden-layer update scale per heartbeat is on the order of only several thousand operations. This is considerably lower than many dense deep-learning models operating on uniformly sampled waveforms, which require repeated convolution or long-sequence processing over dense inputs. At the same time, the proposed model contains only 528 connection weights, and even when biases are included, the total parameter count remains on the order of only a few hundred. These results indicate that, although the proposed method does not surpass the best-performing dense baselines in overall classification accuracy, it is much more compatible with event-driven low-power implementation requirements in terms of input representation, computation pattern, and model compactness.

Overall, the main significance of the proposed time-aware SNN does not lie in outperforming the handcrafted-feature-based framework of Chapter 3 on every metric. Instead, it demonstrates that LC ECG event streams themselves can be directly used for heartbeat classification without relying on manually designed feature construction, and that by explicitly incorporating real inter-event timing into neuron-state updates, the model can effectively exploit the asynchronous temporal structure of the input stream. On this basis, the proposed approach still achieves strong overall classification performance while preserving compact parameter size and sparse event-driven computation. Therefore, from a methodological perspective, the proposed time-aware SNN provides an alternative modeling pathway for LC ECG classification beyond handcrafted feature engineering; from an application perspective, it also offers a more hardware-compatible candidate solution for low-power wearable ECG analysis.

5 Conclusion

This thesis investigated level-crossing (LC)-based event-driven ECG classification for low-power wearable cardiac monitoring applications. Motivated by the limitations of conventionally sampled ECG signals in terms of data redundancy, transmission burden, and continuous digital-processing cost, the study focused on three tightly connected aspects: LC ECG representation, handcrafted-feature-based classification, and event-driven neural modeling. The overall objective was to explore an ECG classification framework that can preserve strong classification capability while remaining compatible with low-power implementation requirements.

First, this thesis established the event-driven representation basis of LC ECG. Unlike conventional uniformly sampled ECG signals, LC representation does not record the entire waveform at fixed time intervals, but instead generates asynchronous crossing events whenever the signal crosses predefined quantization levels. In this way, the dynamic behavior of the ECG waveform can be preserved in a much sparser form. The thesis analyzed the basic LC mechanism and examined how key LC parameters affect signal fidelity and event density, thereby providing a unified signal-representation foundation for subsequent classification methods. The corresponding analysis showed that LC ECG can substantially reduce redundant samples while still retaining the local dynamic information required for heartbeat discrimination, which makes it a promising event-driven input format for low-power ECG intelligence.

On this basis, a handcrafted-feature-based LC ECG classification framework was first developed. This method constructed rhythm features, morphological features, and LC dynamic features from the LC event sequence, and then used a lightweight multilayer perceptron (MLP) as the classifier. Experimental results demonstrated that, under the current experimental setting, this framework achieved strong overall heartbeat classification performance. This result confirms that even under the sparse LC representation, carefully designed features can still support highly effective arrhythmia recognition. It also verifies the validity of LC ECG as an input representation for classification and shows that explicitly designed, physiologically meaningful features still remain highly useful in ECG analysis.

However, although the handcrafted-feature-based approach produced strong results, it still relied on prior knowledge and manual feature engineering. To address this limitation, this thesis further proposed a time-aware spiking neural network (SNN) method that directly models LC event streams. Instead of constructing explicit discriminative feature vectors, the proposed method first represented local event dynamics through the inter-event interval and crossing direction, then transformed these local dynamics into a sequence of discrete slope tokens, and finally encoded the token sequence into multi-channel spike inputs suitable for SNN processing. On top of this input representation, a time-aware SNN classifier was constructed, in which the real temporal interval between neighboring events explicitly participated in neuron-state evolution.

Experimental results showed that the proposed time-aware SNN can directly learn from LC event streams without relying on handcrafted feature construction, while still achieving strong Accuracy and Macro-F1 performance. Compared with a fixed-step SNN, the proposed time-aware mechanism makes better use of the asynchronous timing information embedded in LC ECG, thereby improving the temporal modeling capability of the event-driven classifier. Parameter analysis further showed that hidden-layer size, membrane time constant, and firing threshold all significantly affect both classification performance and neural activity, and that an appropriate parameter configuration enables a favorable balance between predictive performance and event-driven sparsity. Although the overall performance of the proposed time-aware SNN remains slightly below that of the best handcrafted-feature-based method under the current setting, it exhibits clear advantages in model compactness, event-driven computation form, and low-power implementation compatibility.

Taken together, the results of this thesis indicate that LC ECG is not only suitable for high-performance classification through handcrafted feature extraction, but can also serve directly as the input to an event-driven neural model. In other words, LC representation can support both an interpretable feature-engineering pipeline and a more automated event-driven learning pipeline. The handcrafted-feature-based approach demonstrated the effectiveness of LC ECG in explicit feature construction, while the time-aware SNN further demonstrated that the LC event stream itself contains sufficient information for direct learning. In this way, the thesis established a complete research path from event-driven signal representation, to handcrafted-feature classification, and finally to time-aware event-driven neural modeling.

Therefore, the main contributions of this thesis can be summarized as follows. First, it analyzed the event-driven characteristics of LC ECG and verified its feasibility as a low-redundancy input representation for ECG classification. Second, it proposed and validated a handcrafted-feature-based LC ECG classification framework, showing that strong performance can still be achieved under sparse LC representation. Third, it proposed a time-aware SNN framework for LC event streams, in which real inter-event intervals are explicitly incorporated into neuron dynamics for direct asynchronous-event modeling. Fourth, through experimental comparison, it demonstrated that the proposed time-aware SNN not only maintains strong classification capability, but also better aligns event-driven input representation with event-driven neural computation, thereby providing a modeling pathway that is more compatible with low-power wearable ECG intelligence and neuromorphic implementation logic.

5.1 Future Work

Although this thesis has verified the feasibility and effectiveness of LC ECG-based event-driven classification, several directions remain open for further study.

First, the current input encoding of the time-aware SNN still involves a certain amount of manual design. In this thesis, local event dynamics were represented by discrete slope tokens derived from inter-event intervals and crossing directions, and then converted into multi-channel spike inputs. While this design provides a natural interface for SNN processing, it does not yet constitute fully end-to-end learning from raw event streams. Future work may explore more direct event-representation-learning methods, for example by learning event embeddings directly from the raw LC crossing sequence, so that the dependence on manually designed discretization and token definitions can be reduced.

Second, the current training framework is based on surrogate gradients and Backpropagation Through Time (BPTT). Although this supervised training strategy is effective for optimizing classification performance, it introduces relatively high training complexity and remains sensitive to hyperparameter settings. Future work may investigate more efficient learning strategies for event-driven systems, such as online learning mechanisms, hybrid training schemes that combine local plasticity with global supervision, or training-inference co-design frameworks that are more compatible with neuromorphic implementations. Such efforts may help further reduce training cost while preserving or even improving classification capability.

Third, although low-power implementation potential has been emphasized throughout the proposed framework, the present complexity analysis is still mainly limited to algorithm-level indicators such as parameter count, firing rate, and approximate synaptic operations. Future work should extend the analysis to concrete hardware platforms, including FPGA, ASIC, or neuromorphic processors, so that storage overhead, energy consumption, and latency can be evaluated more realistically. This would provide stronger system-level evidence for the practical deployment value of the proposed approach in wearable ECG devices.

Fourth, the experiments in this thesis were mainly conducted on public ECG datasets under heartbeat-level classification settings. Future studies may extend the framework to more challenging application scenarios, such as cross-subject generalization, continuous long-term ECG stream analysis, multi-lead ECG input, and integration with anomaly detection, early warning, or online patient adaptation. Such extensions would not only help assess the robustness of the proposed method under more realistic clinical conditions, but would also move event-driven ECG analysis closer to practical deployment.

Finally, from a broader perspective, the combination of LC representation and event-driven neural processing may be applicable not only to ECG, but also to other physiological signals such as PPG, EEG, or multimodal wearable sensing data. If future work can further unify event-driven front-end acquisition, sparse event-based signal representation, neuromorphic back-end computation, and on-chip intelligent deployment, then a more general and efficient implementation paradigm may emerge for low-power wearable health-monitoring systems.

In summary, this thesis has systematically investigated LC ECG representation and classification from the perspectives of event-driven signal processing and event-driven learning. The results confirm the application potential of LC ECG in low-power wearable cardiac intelligence. Future work will continue toward more end-to-end event representation learning, more efficient training mechanisms, and more hardware-oriented deployment, so as to further improve both the performance and the practicality of event-driven ECG analysis in real-world scenarios.

Bibliography

[1] World Health Organization, "Cardiovascular diseases (cvds)." https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds), 2025. Accessed: 2026-03-20.

[2] Our World in Data, "Share of deaths from cardiovascular diseases." https://ourworldindata.org/grapher/share-deaths-cardiovascular-disease?country= CANmapSelect= CAN, 2024. Accessed: 2026-03-20.

[3] Public Health Agency of Canada, "Heart disease: Monitoring." https://www.canada.ca/en/public-health/services/diseases/heart-health/heart-diseases-conditions/surveillance-heart-diseases-conditions.html, 2026. Government of Canada, Accessed: 2026-03-20.

[4] Heart and Stroke Foundation of Canada, "Connected by the numbers." https://www.heartandstroke.ca/articles/connected-by-the-numbers, 2026. Accessed: 2026-03-20.

[5] The Cardiovascular, "Ecg interpretation: Characteristics of the normal ecg (p-wave, qrs complex, st segment, t-wave)." https://ecgwaves.com/topic/ecg-normal-p-wave-qrs-complex-st-segment-t-wave-j-point/, n.d. Accessed: 2026-03-20.

[6] Wikimedia Commons, "File: Annotated ecgcolor.svg." https://commons.wikimedia.org/wiki/File:Annotated$_{E}CGcolor.s$
$2026 - 03 - 20$.

[7] MedlinePlus, "Holter heart monitor." https://medlineplus.gov/ency/imagepages/8810.htm, n.d. U.S. National Library of Medicine, Accessed: 2026-03-20.

[8] B. Khan, R. T. Khalid, M. H. Masrur, M. Awais, B. Khan, and B. L. Khoo, "Next-generation wearable ecg systems: Soft materials, ai integration, and personalized healthcare applications," *Chemical Engineering Journal*, vol. 525, p. 170117, 2025.

[9] K. Bregar, T. Krivstofelc, M. Depolli, V. Avbelj, and A. Rashkovska, "Power autonomy estimation of low-power sensor for long-term ecg monitoring," *Sensors*, vol. 22, no. 14, p. 5070, 2022.

[10] A. A. A. Abdou and S. K. S. Krishnan, "Horizons in single-lead ecg analysis from devices to data," *Frontiers in Signal Processing*, vol. 2, p. 866047, 2022.

[11] H. Choi, J. Park, J. Lee, and D. Sim, "Review on spiking neural network-based ecg classification methods for low-power environments," *Biomedical Engineering Letters*, vol. 14, no. 4, pp. 849–870, 2024.

[12] X. An, S. Shi, Q. Wang, Y. Yu, and Q. Liu, "Research on a lightweight arrhythmia classification model based on knowledge distillation for wearable single-lead ecg monitoring systems," *Sensors*, vol. 24, no. 24, p. 7896, 2024.

[13] M. Rahman and B. I. Morshed, "Resource-constrained on-chip ai classifier for beat-by-beat real-time arrhythmia detection with an ecg wearable system," *Electronics*, vol. 14, no. 13, p. 2654, 2025.

[14] X. Xu, Q. Cai, Y. Zhao, G. Wang, L. Zhao, and Y. Lian, "A 2.66 μw clinician-like cardiac arrhythmia watchdog based on p-qrs-t for wearable applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 16, no. 5, pp. 793–806, 2022.

[15] Y. Ansari, O. Mourad, K. Qaraqe, and E. Serpedin, "Deep learning for ECG arrhythmia detection and classification: an overview of progress for period 2017–2023," *Frontiers in Physiology*, vol. 14, p. 1246746, 2023.

[16] Q. Xiao, K. Lee, S. A. Mokhtar, I. Ismail, A. L. bin Md Pauzi, Q. Zhang, and P. Y. Lim, "Deep learning-based ECG arrhythmia classification: A systematic review," *Applied Sciences*, vol. 13, no. 8, p. 4964, 2023.

[17] R. Jahangir, M. N. Islam, M. S. Islam, and M. M. Islam, "ECG-based heart arrhythmia classification using feature engineering and a hybrid stacked machine learning," *BMC Cardiovascular Disorders*, vol. 25, p. 146, 2025.

[18] N. Alamatsaz, L. Tabatabaei, M. Yazdchi, H. Payan, N. Alamatsaz, and F. Nasimi, "A lightweight hybrid CNN-LSTM explainable model for ECG-based arrhythmia detection," *Biomedical Signal Processing and Control*, vol. 89, p. 105884, 2024.

[19] H. El-Ghaish and E. Eldele, "ECGTransForm: Empowering adaptive ECG arrhythmia classification framework with bidirectional transformer," *Biomedical Signal Processing and Control*, vol. 89, p. 105714, 2024.

[20] J. Hua, B. Chu, J. Zou, and J. Jia, "ECG signal classification in wearable devices based on compressed domain," *PLOS ONE*, vol. 18, no. 4, p. e0284008, 2023.

[21] C. Zhang, J. Li, P. Guo, Q. Li, X. Zhang, and X. Wang, "A configurable hardware-efficient ECG classification inference engine based on CNN for mobile healthcare applications," *Microelectronics Journal*, vol. 141, p. 105969, 2023.

[22] Y. Liu, J. Liu, Y. Tian, Y. Jin, Z. Li, L. Zhao, and C. Liu, "Pruned lightweight neural networks for arrhythmia classification with clinical 12-lead ECGs," *Applied Soft Computing*, vol. 154, p. 111340, 2024.

[23] P. Busia, M. A. Scrugli, V. J.-B. Jung, L. Benini, and P. Meloni, "A tiny transformer for low-power arrhythmia classification on microcontrollers," *IEEE Transactions on Biomedical Circuits and Systems*, 2024. Early Access.

[24] M. M. Farag, "A tiny matched filter-based CNN for inter-patient ECG classification and arrhythmia detection at the edge," *Sensors*, vol. 23, no. 3, p. 1365, 2023.

[25] X. Xu, Y. Suo, Y. Zhao, P. Zhou, X. Han, Q. Cai, M. Wang, J. Yuan, L. Zhao, Y. Li, G. Wang, and Y. Lian, "A dry-electrode enabled ECG-on-chip with arrhythmia-aware data transmission," *Science China Information Sciences*, vol. 68, p. 122405, 2025.

[26] M. Saeed, Q. Wang, O. Märtens, B. Larras, A. Frappé, B. Cardiff, and D. John, "Evaluation of level-crossing adcs for event-driven ecg classification," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 15, no. 6, pp. 1129–1139, 2021.

[27] Y. Zigel, A. Cohen, and A. Katz, "The weighted diagnostic distortion (wdd) measure for ecg signal compression," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 11, pp. 1422–1430, 2000.

[28] N. Ravanshad, H. Rezaee-Dehsorkh, R. Lotfi, and Y. Lian, "A level-crossing based qrs-detection algorithm for wearable ecg sensors," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 1, pp. 183–192, 2014.

[29] Y. Zhao, Z. Shang, and Y. Lian, "A 13.34 w event-driven patient-specific ann cardiac arrhythmia classifier for wearable ecg sensors," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 2, pp. 186–197, 2020.

[30] M. Fira, L. Goraș, L. Fira, R. F. Popa, and H.-N. Costin, "Arrhythmia classification with single-channel features extracted from "a large-scale 12-lead ecg database for arrhythmia study"," *Sensors*, vol. 25, no. 18, 2025.

[31] "ECG-based heart arrhythmia classification using feature engineering and a hybrid stacked machine learning - BMC Cardiovascular Disorders — link.springer.com." https://link.springer.com/article/10.1186/s12872-025-04678-9?utm_source=chatgpt.com#citeas. [Accessed 28-03-2026].

[32] J. Duforest, B. Larras, A. Frappé, C. J. Deepu, and O. Märtens, "Antidictionary-based cardiac arrhythmia classification for smart ecg sensors," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 414–418, 2022.

[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[34] Z. Wu and C. Guo, "Deep learning and electrocardiography: systematic review of current techniques in cardiovascular disease diagnosis and management," *BioMedical Engineering OnLine*, vol. 24, no. 1, p. 23, 2025.

[35] Y. Li *et al.*, "Deep learning-based ecg arrhythmia classification: A systematic review," *Applied Sciences*, vol. 13, no. 8, p. 4964, 2023.

[36] Y. Yang *et al.*, "Resource-constrained on-chip ai classifier for beat-by-beat real-time ecg arrhythmia classification," *Electronics*, vol. 14, no. 13, p. 2654, 2025.

[37] S. Rani, D. Gopal, S. Vignesh, and P. Ramesh, "Advanced iot-enabled embedded wearable system for continuous cardiovascular health monitoring using edge intelligence," *Biomedical Signal Processing and Control*, vol. 111, p. 108362, 2026.

[38] C. Zhou, H. Zhang, L. Yu, Y. Ye, Z. Zhou, L. Huang, Z. Ma, X. Fan, H. Zhou, and Y. Tian, "Direct training high-performance deep spiking neural networks: A review of theories and methods," *Frontiers in Neuroscience*, vol. 18, p. 1383844, 2024.

[39] J. Wu, Y. Wang, Z. Li, L. Lu, and Q. Li, "A review of computing with spiking neural networks," *Computers, Materials and Continua*, vol. 78, no. 3, pp. 2909–2939, 2024.

[40] Y. Xing, L. Zhang, Z. Hou, X. Li, Y. Shi, Y. Yuan, F. Zhang, S. Liang, Z. Li, and L. Yan, "Accurate ecg classification based on spiking neural network and attentional mechanism for real-time implementation on personal portable devices," *Electronics*, vol. 11, no. 12, 2022.

[41] H. Chu, Y. Yan, L. Gan, H. Jia, L. Qian, Y. Huan, L. Zheng, and Z. Zou, "A neuromorphic processing system with spike-driven snn processor for wearable ecg classification," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 16, no. 4, pp. 511–523, 2022.

[42] F. Jiang, J. Xiao, L. Liu, and C. Wang, "Dceten: A lightweight ecg automatic classification network based on transformer model," *Digital Communications and Networks*, 2024.

[43] M. Rahman and B. I. Morshed, "Resource-constrained on-chip ai classifier for beat-by-beat real-time arrhythmia detection with an ecg wearable system," *Electronics*, vol. 14, no. 13, 2025.