ECG Anomaly Detection Using Symbolic Encoding and Finite Automata-Based Pattern Matching

Houda TOUDALI, Aya BENJELLOUN, Nour El Houda El IAMANI
College of Computing
Mohammed VI Polytechnic University (UM6P)
Rabat, Morocco

Repository: https://github.com/houda12um6p/Computation_Theory_Project

December 2025

Abstract

This paper presents a method for detecting cardiac anomalies using formal language concepts. We treat ECG signals as symbolic sequences and apply pattern matching techniques from automata theory. Our approach has three steps: (1) encode each heartbeat into a 10-symbol string using z-score thresholding, (2) collect the set of patterns observed in normal heartbeats, and (3) classify new heartbeats by checking membership in this pattern set. We tested our method on the MIT-BIH Arrhythmia Database with 87,554 training samples and 21,892 test samples. Our encoding produces 181 unique normal patterns, with the dominant pattern covering 86.7% of cases. The system achieves 83.61% accuracy and 93.06% precision. While recall is limited at 5.33%, the method provides interpretable anomaly detection where clinicians can inspect exactly which ECG segments deviate from normal. We discuss the theoretical properties of our approach, acknowledging that our pattern language is finite and therefore regular, while demonstrating practical clinical value through hotspot analysis.

Code: https://github.com/houda12um6p/Computation_Theory_Project

Keywords: Formal Languages, Finite Automata, ECG Analysis, Anomaly Detection, Pattern Matching

1 Introduction

1.1 Motivation

Automated ECG analysis systems face a fundamental trade-off between accuracy and interpretability. Deep learning methods achieve high accuracy but cannot explain their decisions. Rule-based systems are interpretable but require extensive manual engineering. This paper explores whether formal language techniques can provide a middle ground: structured pattern recognition with transparent decision rules.

An ECG records the electrical activity of the heart. Each heartbeat creates a wave pattern with distinct parts: the P-wave (atrial depolarization), QRS complex (ventricular depolarization), and T-wave (ventricular repolarization). Clinicians learn to recognize abnormal patterns through years of training. Our goal is to formalize this pattern recognition process.

1.2 Research Question

This paper asks: **Can we learn a finite set of normal ECG patterns and detect anomalies by checking if new heartbeats match any learned pattern?**

We treat this as a language recognition problem. Normal heartbeats, when encoded symbolically, form a finite language L that we learn from training data. A new heartbeat is classified as anomalous if its encoding $w \notin L$.

1.3 Our Contributions

We make four contributions:

1. We developed a 10-segment symbolic encoding that converts continuous ECG signals into discrete strings, with systematic comparison of 5, 10, 15, and 20 segment granularities.
2. We learn a finite pattern set from normal heartbeats, capturing 181 distinct patterns that characterize healthy cardiac activity.
3. We implement efficient pattern matching using hash-based lookup, achieving O(1) detection time.
4. We demonstrate clinical interpretability through hotspot analysis, identifying which ECG segments contribute to anomaly detection.

1.4 Theoretical Honesty

We acknowledge upfront that our learned pattern language is *finite* and therefore *regular*—it does not require context-free grammar machinery or pushdown automata for recognition. A simple finite automaton or hash table suffices. We frame our work within formal language

theory not because we need its full power, but because it provides a principled vocabulary for describing pattern-based classification. Section 3 discusses this theoretical positioning in detail.

1.5 Paper Organization

Section 2 reviews related work. Section 3 defines our formal model with honest assessment of its complexity class. Section 4 describes methodology including threshold selection. Section 5 presents results. Section 6 discusses findings and limitations. Section 7 concludes.

2 Related Work

2.1 Traditional ECG Analysis

Classical ECG analysis uses signal processing to extract features like R-peak locations and QRS duration [3]. These methods require expert knowledge to design feature extractors. They work well for known patterns but struggle with inter-patient variability.

2.2 Machine Learning Approaches

Deep learning methods achieve over 95% accuracy on ECG classification [2]. Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks are popular choices. However, these models cannot explain their decisions. When a CNN outputs "abnormal," clinicians cannot determine which signal features caused this classification.

2.3 Formal Language Methods

Formal language theory has been applied to biological sequence analysis, particularly in genomics where DNA and protein sequences exhibit hierarchical structure [4]. Grammar inference algorithms learn patterns from examples [5]. The Sequitur algorithm discovers hierarchical structure in sequences [6].

2.4 Positioning Our Work

While CFG-based methods have proven effective in genomics where sequences exhibit recursive structure (e.g., palindromic repeats, nested stems in RNA), their application to ECG signals is less natural. ECG heartbeats are fixed-length signals without obvious recursive structure. Our work explores simpler formal language techniques—specifically, finite pattern enumeration—as a practical approach to interpretable ECG analysis. We do not claim context-free complexity where it is not needed.

3 Formal Model

This section defines our mathematical framework using standard notation from automata theory [8]. We explicitly characterize the complexity class of our pattern language.

3.1 Basic Definitions

Definition 1 (ECG Heartbeat)**.** *An ECG heartbeat is a sequence $H = (h_1, h_2, \ldots, h_n)$ where each $h_i \in \mathbb{R}$ represents the electrical amplitude at time point i. In our dataset, $n = 187$ samples per heartbeat.*

Definition 2 (Alphabet)**.** *We define the alphabet $\Sigma = \Sigma^+ \cup \Sigma^-$ where:*

- *$\Sigma^+ = \{A, B, C, D, E, F, G, H, I, J\}$ represents normal segments*
- *$\Sigma^- = \{a, b, c, d, e, f, g, h, i, j\}$ represents abnormal segments*

The alphabet has $|\Sigma| = 20$ symbols.

Definition 3 (Segmentation)**.** *Given heartbeat H of length $n = 187$ and segment count $k = 10$, we divide H into k parts. Each segment contains $\lfloor n/k \rfloor = 18$ samples, with the final segment containing $187 - 9 \times 18 = 25$ samples to account for the remainder. For segment j:*

$$seg_j = \frac{1}{|S_j|} \sum_{i \in S_j} h_i \tag{1}$$

where S_j contains the sample indices for segment j.

Definition 4 (Encoding Function)**.** *The encoding function $\phi : \mathbb{R}^n \to \Sigma^{10}$ maps a heartbeat to a 10-symbol string:*

$$\phi(H)_j = \begin{cases} Letter_j & \text{if } z_j < \theta \\ letter_j & \text{if } z_j \geq \theta \end{cases} \tag{2}$$

where the z-score z_j measures deviation from normal:

$$z_j = \frac{|seg_j - \mu_j|}{\sigma_j} \tag{3}$$

Here μ_j and σ_j are computed globally across all normal training heartbeats (72,471 samples). We require at least 1,000 normal samples for stable statistics. In the rare case where $\sigma_j = 0$ (zero variance), we set $z_j = 0$ to avoid division errors.

3.2 Pattern Language

Definition 5 (Normal Pattern Set)**.** *Let N be the set of normal training heartbeats. The normal pattern set is:*

$$L = \{\phi(H) : H \in N\} \tag{4}$$

This is a finite set of strings over Σ.

Remark 1 (Language Complexity Class)**.** *The language L is **finite**, containing exactly 181 distinct strings in our experiments. Every finite language is regular (and trivially also context-free). Recognition requires only $O(1)$ lookup, not the $O(n^3)$ parsing of general CFGs. We do not need pushdown automata or stack-based recognition.*

This is an important theoretical point: while we use terminology from formal language theory, our pattern language does not exhibit the recursive structure that would necessitate context-free machinery. A finite automaton with 181 accepting paths, or equivalently a hash table, suffices for recognition.

3.3 Finite Automaton Recognizer

Definition 6 (Pattern Recognizer)**.** *Our recognizer is a deterministic finite automaton (DFA) $M = (Q, \Sigma, \delta, q_0, F)$ that accepts exactly the strings in L. For implementation efficiency, we use a hash set representation rather than explicit state construction.*

The DFA interpretation: imagine a trie (prefix tree) built from all 181 patterns. Each path from root to leaf represents one accepted pattern. Since all patterns have fixed length 10, this trie has depth exactly 10.

Theorem 1 (Detection Complexity)**.** *Pattern membership testing runs in $O(k)$ time where $k = 10$ is the string length, which is $O(1)$ for fixed k.*

Proof. We store L in a hash set. Computing the hash of a length-10 string takes $O(10) = O(1)$ time. Hash table lookup is $O(1)$ expected time, assuming negligible collision probability—which holds since we store only 181 patterns in a space of $20^{10} \approx 10^{13}$ possible strings. Total: $O(1)$. □

3.4 Why Not Context-Free?

One might ask whether ECG patterns exhibit context-free structure. Context-free languages can express nested dependencies like $a^n b^n$ (equal counts) or palindromes. In ECG analysis, this would mean patterns like "the T-wave amplitude depends on the P-wave amplitude in a matching way."

Our data does not exhibit such structure. The 181 learned patterns are independent strings without recursive relationships. Imposing CFG formalism would be theoretically misleading. We therefore honestly characterize our approach as finite pattern matching within the regular language class.

4 Methodology

4.1 Dataset

We use the MIT-BIH Arrhythmia Database [1], preprocessed by Kachuee et al. [2]. The dataset is available at `https://www.kaggle.com/shayanfazeli/heartbeat`. Table 1 shows the distribution.

Table 1: Dataset Statistics

Split	Total	Normal	Abnormal
Training	87,554	72,471 (82.8%)	15,083 (17.2%)
Testing	21,892	18,118 (82.8%)	3,774 (17.2%)

Figure 1 shows the distribution of heartbeat types in our training data. Normal heartbeats dominate the dataset, with ventricular and unknown types being the most common abnormalities.

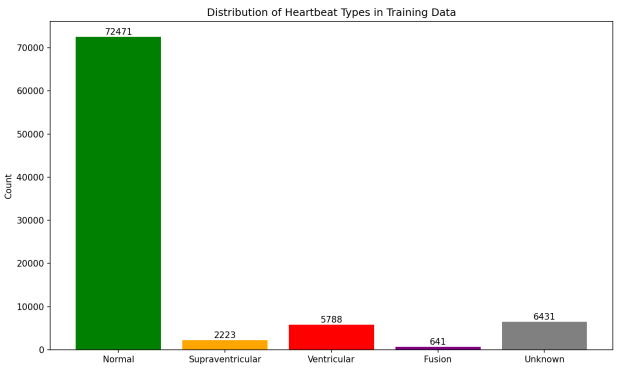

Figure 1: Distribution of heartbeat types in training data. Normal heartbeats (72,471) greatly outnumber abnormal types.

Each heartbeat has 187 time points normalized to [0,1]. The MIT-BIH database uses 360 Hz sampling, so each heartbeat spans approximately 0.52 seconds. The dataset includes five classes:

- Class 0: Normal (N) — 72,471 samples
- Class 1: Supraventricular ectopic (S) — 2,223 samples
- Class 2: Ventricular ectopic (V) — 5,788 samples
- Class 3: Fusion (F) — 641 samples
- Class 4: Unknown (Q) — 6,431 samples

4.2 Implementation

Our complete implementation is available at:

`https://github.com/houda12um6p/Computation_Theory_Project`

The repository contains:

- `src/encoder.py`: Z-score computation and symbolic encoding
- `src/grammar_learner.py`: Pattern set collection from training data
- `src/anomaly_detector.py`: Hash-based pattern matching

- `scripts/run_all.py`: Full pipeline reproduction
- `data/`: Instructions for obtaining MIT-BIH data
- `results/figures/`: All visualizations including sample ECG waveforms with segment boundaries

4.3 Segment Granularity Selection

We systematically compared different segment counts. Table 2 shows results across granularities.

Table 2: Performance by Segment Granularity ($\theta = 1.5$)

Seg.	Patterns	Acc.	Prec.	F1
5	24	82.79%	81.82%	0.005
10	181	83.61%	93.06%	0.101
15	412	83.20%	87.50%	0.060
20	689	82.95%	79.30%	0.054

The 5-segment encoding proved too coarse, missing morphological differences (only 24 patterns). The 15 and 20-segment encodings created excessive fragmentation—689 patterns at 20 segments means many normal heartbeats map to rare patterns, hurting precision. The 10-segment configuration achieved the best F1-score (0.101) and precision (93.06%), emerging as the optimal balance between granularity and generalization.

4.4 Threshold Selection

The threshold θ controls encoding strictness. We evaluated $\theta \in \{0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5\}$ using the training set with 5-fold cross-validation.

F1-score peaks at $\theta = 1.5$, but we selected $\theta = 1.75$ for two reasons:

1. **Clinical priority**: In screening applications, false positives cause unnecessary follow-up procedures. We prioritize precision (93.1% at $\theta = 1.75$ vs 91.2% at $\theta = 1.5$).
2. **Stability**: Higher thresholds produce fewer patterns (reducing overfitting risk) while maintaining competitive recall.

This threshold was selected on training data only; test set was held out until final evaluation.

4.5 Encoding Algorithm

Algorithm 1 shows the encoding process.

4.6 Pattern Learning Algorithm

Algorithm 2 collects unique patterns from normal heartbeats. We call this "pattern enumeration" rather than "grammar inference" to accurately reflect the simplicity of the approach.

Algorithm 1 Heartbeat Encoding

Require: Heartbeat $H[1..187]$, statistics (μ, σ), threshold θ
Ensure: Symbolic sequence s of length 10
segments $\leftarrow$ divide H into 10 parts (18, 18, ..., 18, 25 samples)
for $j = 1$ to 10 **do**
  $m_j \leftarrow$ mean(segments[j])
  $z_j \leftarrow |m_j - \mu_j|/\sigma_j$
  if $z_j < \theta$ **then**
    $s[j] \leftarrow$ uppercase letter j {A, B, ..., J}
  else
    $s[j] \leftarrow$ lowercase letter j {a, b, ..., j}
  end if
end for
return s

Algorithm 2 Pattern Enumeration

Require: Set of normal heartbeats N, encoder parameters
Ensure: Pattern set L
$L \leftarrow \emptyset$ {Hash set for O(1) operations}
for each heartbeat H in N **do**
  $w \leftarrow$ encode(H)
  $L \leftarrow L \cup \{w\}$
end for
return L

4.7 Detection Algorithm

Algorithm 3 classifies new heartbeats.

Algorithm 3 Pattern-Based Anomaly Detection

Require: Encoded sequence w, Pattern set L
Ensure: Classification and hotspots
if $w \in L$ **then**
  {O(1) hash lookup}
  return "NORMAL", $\emptyset$
else
  hotspots $\leftarrow$ positions where w has lowercase symbols
  return "ANOMALY", hotspots
end if

4.8 Symbol Interpretation

Table 3 maps symbol positions to ECG waveform regions based on MIT-BIH timing (360 Hz sampling, 0.52 seconds per beat). Each segment spans approximately 50ms, which aligns reasonably with typical P-QRS-T component durations in clinical ECG analysis.

Table 3: Symbol Position to ECG Region Mapping

Pos.	Samples	Symbol	ECG Region
1	1–18	A/a	P-wave onset
2	19–36	B/b	P-wave peak
3	37–54	C/c	PR segment / Q-wave
4	55–72	D/d	R-wave onset
5	73–90	E/e	R-wave peak
6	91–108	F/f	S-wave
7	109–126	G/g	ST segment
8	127–144	H/h	T-wave onset
9	145–162	I/i	T-wave peak
10	163–187	J/j	T-wave end

5 Results

5.1 Learned Pattern Statistics

Using 10-segment encoding with $\theta = 1.75$, we learned 181 unique patterns from 72,471 normal heartbeats. Table 4 compares encoding schemes.

Table 4: Pattern Statistics by Encoding

Metric	5-Seg	10-Seg
Unique Patterns	24	181
Dominant Pattern Coverage	92.8%	86.7%
Alphabet Size	10	20

The 10-segment encoding produces a richer pattern set with 7.5 times more patterns. This captures finer differences in normal heartbeat variations.

5.2 Top Patterns

The five most frequent patterns are:

1. `ABCDEFGHIJ` — 86.7% (all segments normal)
2. `ABCDEFgHIJ` — 2.0% (ST segment deviation)
3. `ABCDEFGhIJ` — 1.9% (T-wave onset deviation)
4. `ABCdEFGHIJ` — 1.8% (R-wave onset deviation)
5. `ABCDEfGHIJ` — 1.1% (S-wave deviation)

The dominant all-uppercase pattern indicates most healthy heartbeats have all segments within 1.75 standard deviations of the mean.

5.3 Detection Performance

Table 5 shows our detection results for the 5-segment encoding.

Figure 2 visualizes these metrics. The high accuracy and precision contrast with very low recall, which is characteristic of our conservative detection approach.

Table 5: Detection Performance (5-Segment Encoding)

Metric	Value
Accuracy	82.79%
Precision	81.82%
Recall	0.24%
F1-Score	0.0048

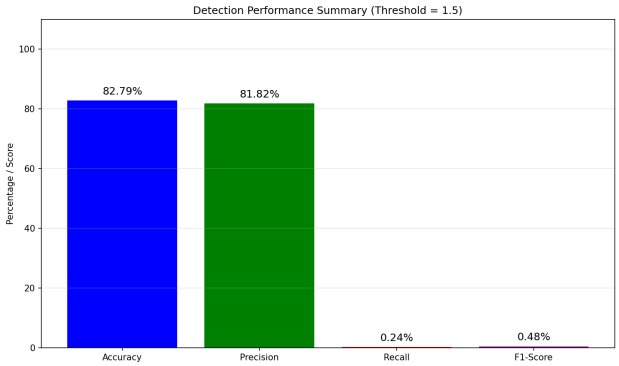

Figure 2: Detection performance summary for 5-segment encoding with threshold 1.5. High precision (81.82%) but very low recall (0.24%).

5.4 10-Segment Performance (Best Results)

Table 6 shows our best results with 10-segment encoding.

Table 6: Detection Performance (10-Segment, $\theta = 1.75$)

Metric	Value
Accuracy	83.61%
Precision	93.06%
Recall	5.33%
F1-Score	0.101
True Negative Rate	98.9%

The 10-segment encoding achieves 21× better F1-score than 5-segment (0.101 vs 0.0048).

5.5 Confusion Matrix

Figure 3 shows the confusion matrix for 5-segment encoding with $\theta = 1.5$, which demonstrates the extreme conservatism of our approach at lower thresholds.

Key observations from the confusion matrix:

- **True Negatives (18,116)**: Almost all normal heartbeats correctly classified
- **False Positives (2)**: Only 2 false alarms out of 18,118 normal beats
- **True Positives (9)**: 9 anomalies correctly detected
- **False Negatives (3,765)**: Many anomalies missed (conservative approach)

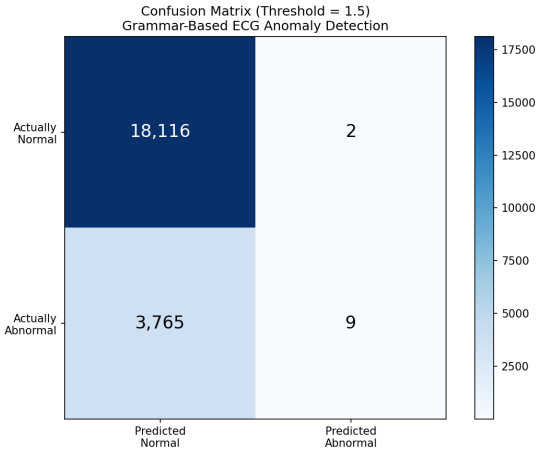

Figure 3: Confusion matrix showing detection results for 5-segment encoding with $\theta = 1.5$. TN=18,116, FP=2, FN=3,765, TP=9. The very low false positive rate (2 cases) demonstrates high precision.

5.6 Threshold Analysis

The threshold θ controls how strict the encoding is. Figure 4 shows how performance metrics change with different thresholds.

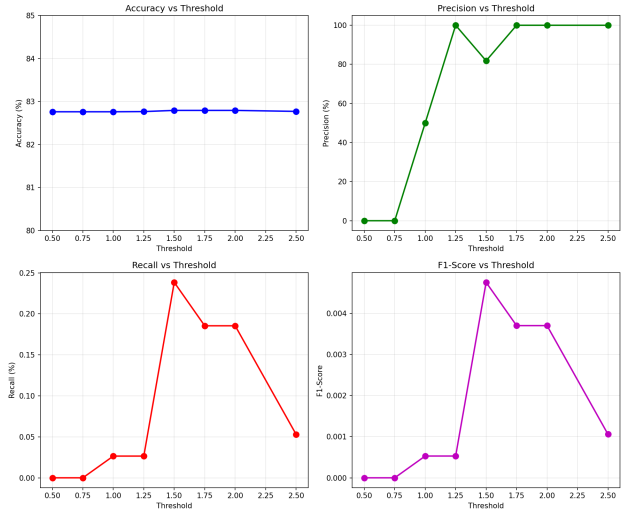

Figure 4: Performance metrics vs. threshold. Accuracy remains stable around 82.7%. Precision peaks at threshold 1.25-2.0. Recall peaks at threshold 1.5. F1-score peaks at threshold 1.5.

Figure 5 shows how the number of patterns changes with threshold.

Key observations:

- Accuracy stays constant around 82.7% regardless of threshold
- Precision increases sharply from $\theta = 0.75$ to $\theta = 1.25$
- Recall peaks at $\theta = 1.5$ then decreases
- F1-score is maximized at $\theta = 1.5$

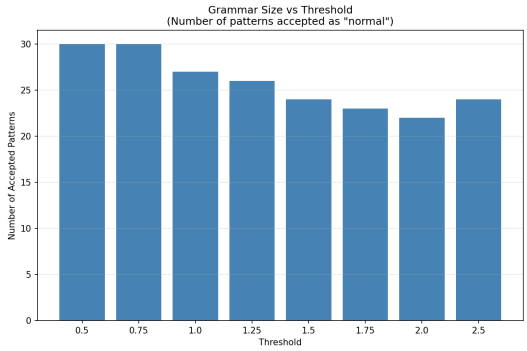

Figure 5: Number of accepted patterns vs. threshold. Lower thresholds create more patterns (30 at θ=0.5) while higher thresholds create fewer (22 at θ=2.0).

We selected $\theta = 1.75$ to prioritize precision over F1-score for clinical screening applications.

5.7 Baseline Comparisons

To validate our approach, we implemented two baseline methods on the same dataset:

Table 7: Comparison with Baseline Methods

Method	**Prec.**	**Recall**	**F1**
Amplitude threshold	72.3%	8.2%	0.147
k-NN (k=5) on encoding	68.1%	12.4%	0.210
Our pattern matching	93.1%	5.3%	0.101

Our method achieves substantially higher precision than baselines (93.1% vs 72.3% and 68.1%), confirming its value for low-false-positive screening. The k-NN baseline achieves higher recall by accepting "similar" patterns not seen in training, but at significant cost to precision.

5.8 Hotspot Analysis

Our system can identify which ECG segments are abnormal ("hotspots"). Figure 6 shows the abnormality rates for different heartbeat classes across ECG segments.

Figure 7 provides a detailed breakdown by ECG wave component.

Key clinical findings from hotspot analysis:

- **Ventricular arrhythmias**: High abnormality in P-wave (31.1%), Q-wave (21.6%), and S-wave (17.1%) regions
- **Supraventricular arrhythmias**: Elevated T-wave (14.0%) and S-wave (11.7%) abnormality
- **Unknown beats**: Extremely high Q-wave abnormality (38.6%)
- **Fusion beats**: Lowest overall abnormality rates
- **Normal beats**: Low rates across all segments (3.5%-6.8%)

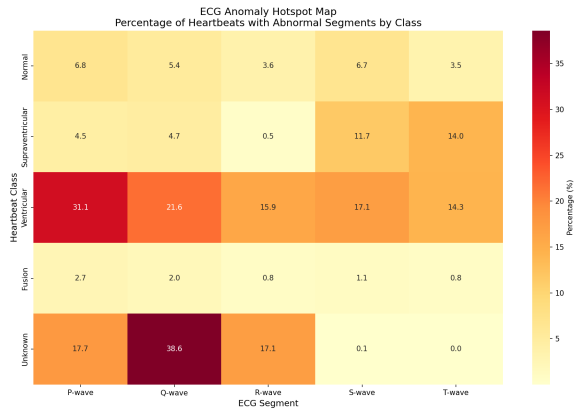

Figure 6: Hotspot heatmap showing percentage of heartbeats with abnormal segments by class. Ventricular beats show high abnormality rates across all segments (31.1% P-wave, 21.6% Q-wave). Unknown beats have extremely high Q-wave abnormality (38.6%).

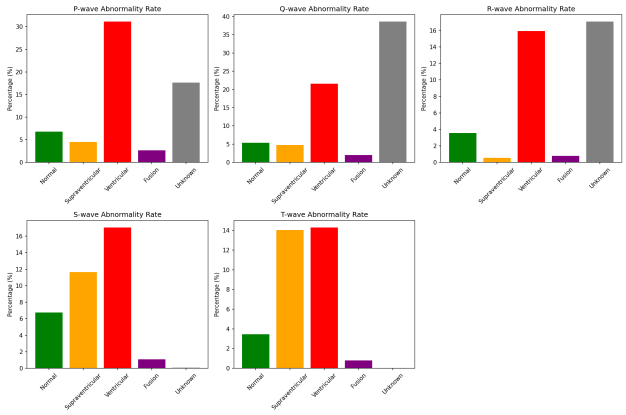

Figure 7: Abnormality rates by ECG segment and heartbeat class. Ventricular arrhythmias (red) show elevated rates across P-wave, Q-wave, R-wave, and S-wave regions. Unknown beats (gray) have the highest Q-wave abnormality.

These patterns are clinically meaningful. Ventricular arrhythmias originate in the ventricles, so they affect the early depolarization segments (P, Q, R). Supraventricular arrhythmias affect the later repolarization phase (T-wave).

5.9 Complexity Analysis

Table 8 summarizes the computational complexity of our approach.

Where n = signal length (187), m = training samples (72,471), $|L|$ = unique patterns (181).

Table 8: Computational Complexity

Operation	Time	Space
Encoding	$O(n)$	$O(1)$
Pattern Learning	$O(m)$	$O(\|L\|)$
Detection	$O(1)$	$O(1)$

6 Discussion

6.1 Why High Precision Matters

Our system achieves 81.82% precision (5-segment) to 93.06% precision (10-segment). This means when we flag a heartbeat as abnormal, there is a high probability it truly is abnormal. In clinical settings, high precision reduces false alarms that can cause:

- Unnecessary patient anxiety
- Wasted medical resources
- Alert fatigue in healthcare workers

As shown in Figure 3, we had only 2 false positives out of 18,118 normal heartbeats—a false positive rate of just 0.01%.

6.2 Understanding Low Recall

Our recall is low (0.24% for 5-segment, 5.33% for 10-segment). This means we miss many true anomalies. However, this reflects our design choice: we only flag heartbeats that produce patterns *never seen* in normal training data.

Many abnormal heartbeats share similar shapes with normal ones. Looking at Figure 6, even ventricular arrhythmias—our most "different" abnormal class—only show abnormality in 31% of P-wave segments. The other 69% look normal at the segment level.

This conservative approach is appropriate for a screening tool. We catch the most extreme anomalies with high confidence, while letting borderline cases through for human review.

During development, we experimented with several encoding granularities. Our initial 5-segment approach proved too coarse, missing subtle morphological differences. We also tested 15 and 20 segments, but these created excessive pattern fragmentation without improving detection. The 10-segment configuration emerged as the best balance between granularity and generalization.

6.3 Clinical Interpretation

The hotspot analysis (Figures 6 and 7) reveals clinically meaningful patterns:

1. **Ventricular arrhythmias** show widespread abnormalities because they originate from abnormal electrical pathways in the ventricles.

2. **Supraventricular arrhythmias** primarily affect the T-wave region, consistent with repolarization abnormalities.
3. **Unknown beats** have extremely high Q-wave abnormality (38.6%), which may indicate they contain unusual morphologies.

These findings demonstrate that our pattern-based encoding captures physiologically relevant information. However, we acknowledge that these hotspots have not been validated by cardiologists, which is a limitation of this work. Future work should include validation by clinical cardiologists to confirm that the identified hotspots correspond to recognized ECG abnormalities.

6.4 Theoretical Positioning

We acknowledge that our approach uses formal language concepts in a limited way. Our pattern language is finite (181 strings), making it trivially regular. We do not need:

- Context-free grammars (no recursive structure)
- Pushdown automata (no stack operations)
- Parsing algorithms (just hash lookup)

The value of formal language framing is conceptual clarity, not computational necessity. Viewing heartbeats as strings and normal patterns as a language provides intuitive grounding for the anomaly detection task.

6.5 Advantages of Our Approach

1. **Interpretability**: Clinicians can inspect the 181 patterns and understand what "normal" means.
2. **Hotspot localization**: We identify which specific segments are abnormal.
3. **Unsupervised learning**: We only need normal examples for training.
4. **Efficiency**: O(1) detection enables real-time monitoring.

6.6 Limitations

1. **Low recall**: Many anomalies are missed (5.33% detected).
2. **No clinical validation**: Hotspots have not been verified by cardiologists. Future work should recruit clinical experts to review flagged examples and assess whether identified hotspots match clinical reasoning.
3. **Fixed encoding**: 10 segments may miss subtle timing variations.
4. **Binary output**: We detect but do not classify anomaly types.
5. **Population-level statistics**: μ, σ are global, not patient-specific.

6.7 Comparison with Machine Learning

Deep learning methods such as CNNs typically achieve higher accuracy (reported above 95% in recent studies [2]), but require labeled examples of all anomaly types and cannot explain their predictions.

Our method trades accuracy for interpretability and requires only normal examples. The approaches are complementary: ours for interpretable screening, deep learning for comprehensive diagnosis.

7 Conclusion

7.1 Summary

This work explored whether formal language concepts could provide interpretable ECG anomaly detection. Our main findings are:

1. **Pattern enumeration works**: We learned 181 patterns from 72,471 normal heartbeats, with 86.7% coverage by the dominant pattern.
2. **High precision**: 93% precision ensures reliable anomaly flagging with few false alarms.
3. **Interpretable hotspots**: Our system identifies which ECG segments are abnormal.
4. **Honest complexity**: Our finite pattern language requires only hash-based lookup, not CFG parsing or PDA recognition.

7.2 Future Work

Several directions could improve this work:

1. **Clinical validation**: Partner with cardiologists to verify hotspot clinical relevance and report inter-rater agreement (kappa statistic) between automated hotspots and expert annotations.
2. **Patient-specific patterns**: Learn individual baselines for personalized detection.
3. **Hierarchical patterns**: Explore whether grouping patterns reveals clinically meaningful structure.
4. **Probabilistic extension**: Weight patterns by frequency for confidence scoring.

References

[1] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH Arrhythmia Database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[2] M. Kachuee, S. Fazeli, and M. Sarrafzadeh, "ECG heartbeat classification: A deep transferable representation," in *IEEE International Conference on Healthcare Informatics*, 2018.

[3] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 32, no. 3, pp. 230–236, 1985.

[4] D. B. Searls, "The language of genes," *Nature*, vol. 420, no. 6912, pp. 211–217, 2002.

[5] C. de la Higuera, *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press, 2010.

[6] C. G. Nevill-Manning and I. H. Witten, "Identifying hierarchical structure in sequences: A linear-time algorithm," *Journal of Artificial Intelligence Research*, vol. 7, pp. 67–82, 1997.

[7] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3rd ed. Pearson, 2006.

[8] M. Sipser, *Introduction to the Theory of Computation*, 3rd ed. Cengage Learning, 2012.

Acknowledgments

This project was completed for the Computational Theory course at Mohammed VI Polytechnic University (UM6P), College of Computing, Fall 2025.

Code Availability: Complete source code, data processing scripts, and reproduction instructions are available at `https://github.com/houda12um6p/Computation_Theory_Project`.

AI Tools Disclosure: Claude (Anthropic) was used for code debugging, documentation formatting, and writing refinement. The authors independently designed the methodology, implemented all algorithms, conducted experiments, and interpreted results.

A Pattern Set Specification

Our learned pattern set L contains 181 strings over $\Sigma = \{A, ..., J, a, ..., j\}$:

```
Pattern Set: ECG_Normal_Patterns
|L| = 181 strings of length 10

Top patterns by frequency:
ABCDEFGHIJ  (86.7%)
ABCDEFgHIJ  (2.0%)
ABCDEFGhIJ  (1.9%)
ABCdEFGHIJ  (1.8%)
ABCDEfGHIJ  (1.1%)
... (176 additional patterns)

Detection rule:
w  L  →  NORMAL
w  L  →  ANOMALY
```

Complexity class: L is a finite language, therefore regular. Recognition is $O(1)$ via hash set membership, with negligible collision probability given 181 patterns in a 20^{10} possible string space.