複数系列データにおける連鎖パタンマイニングに関する研究

A Study on Linkage Pattern Mining in Multiple Sequential Data
(複数系列データにおける連鎖パタンマイニングに関する研究)

室蘭工業大学　工学研究科
生産情報システム工学系専攻　13092502
生命情報工学研究室

Saerom Lee

CONTENTS

1 Introduction

1.1 Background

Sequential pattern mining is a promising and effective data mining method for finding frequent patterns in large-scale sequential data. After Agrawal et al. [1] constructed the foundations of sequential pattern mining in 1995, various new effective algorithms have been developed [2, 3] and applied in a wide range of fields such as web log analysis [4], market basket analysis [5], behavior analysis [6], process analysis [7], and DNA sequence analysis [8]. Research into sequential pattern mining can be broadly classified into two types: approaches that target single sequential data and those that target multiple sequential data. The former aims to find repeating and frequently occurring patterns (frequent patterns or episodes) in sequential data [9–13]. The latter focuses on detecting same or similar subsequences among sequential data [14–16].

1.2 Previous Work

Recently, Miura and Okada [17] proposed a method for mining a linkage pattern that is a set of patterns that repeats across multiple sequential data. In their method, linkage patterns were extracted using an interval graph representation of frequent patterns in the sequential data. Note that linkage pattern mining does not assume similarity or correlation among different sequential data patterns. Figure.1.1 shows an example of a linkage pattern A, B, C that appears across three sets of sequential data. As we can see, even if patterns that occur frequently in the respective sequential data do not show similarity to each other, the set of those patterns is extracted as a linkage pattern if it appears continually within the same period. Miura ' s method demonstrated good performance on sequential data without noise/fluctuations [17]; however, they suggested that noise/fluctuations within the sequential data can significantly affect the accuracy of extracting linkage patterns.

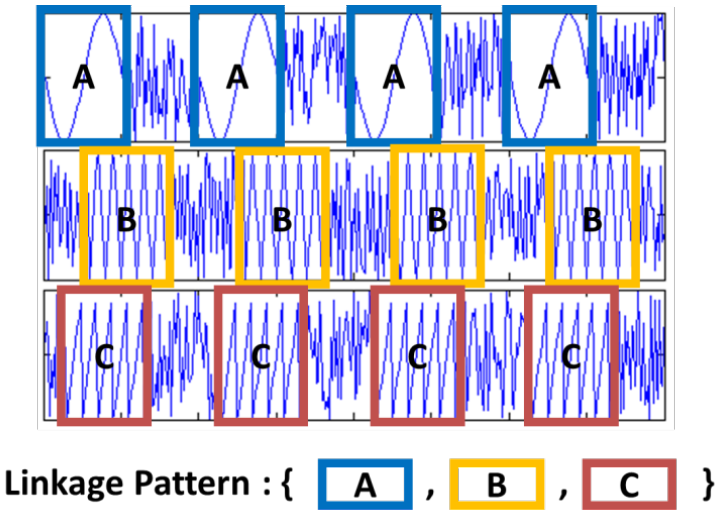


Figure. 1.1: Linkage pattern repeating across three sequential data

1.3 Objective

1.3.1 Linkage Pattern Mining in Multiple Sequential Data

This study develops a noise-robust linkage pattern mining method by improving Miura's method. In our method, closed itemset mining is employed to exclude pseudo patterns generated by noise/fluctuations and obtain only frequent and maximal patterns among different interval graphs. In this study, comparative performance results between the proposed method and Miura's method (hereafter referred to as "the previous method") are shown using artificial sequential datasets.

Definition of Linkage Pattern

Let S be a single sequential data. $freq(S, \alpha)$ is the number of occurrences of a subsequence α in S. For a pre-defined constant value θ, α is a frequent pattern in S if $freq(S, \alpha) \geq \theta$. Suppose that multiple sequential data are given as input, and that frequent patterns have already been extracted from those sequential data. If frequent patterns occurring over those sequential data in a certain time frame satisfy the following two conditions, the set of those frequent patterns is called a linkage pattern.

1) For all the frequent patterns, there exist one or more frequent patterns whose occurring time zones overlap partially or entirely with each other.

2) A set of the frequent patterns that satisfy condition 1) occurs x or more times along the sequential data.

1.3.2 Application to ECG

ECG (electrocardiogram) is used for applying real data to proposed method. However, it is possible to lose important information for detection of abnormal waveform of ECG in the step of discretization of proposed method. Because, amplitude of wave on ECG changing drastically. Because of above reason, we proposed a new discretization for application to ECG. In the experiment, the new method is evaluated for extraction accuracy using real healthy/disease ECG. Furthermore, comparative performance results between the new method and proposed method are shown for practicability of new method.

1.4 Paper Organization

The remainder of this paper is organized as follows. Chapter1 introduces related work and explains objective of this study. Chapter 2 explains previous and proposed method and presents the experimental results using artificial data. Chapter 3 explains application to ECG and presents the experimental results using disease ECG data.

2 Linkage Pattern Mining in Multiple Sequential Data

Figure.2.1 shows the procedure of the proposed method. Figure.2.1a, 2.1b, and 2.1d are the steps implemented in the previous method: extracting and labeling frequent patterns from each sequence (Figure.2.1a), generating interval graphs depending on overlapping labels on the time axis (Figure.2.1b), and outputting the linkage pattern (Figure.2.1d). In this method, a new step (Figure.2.1c) is introduced, i.e., closed itemset mining from the generated interval graphs. This resolves the problem by which linkage patterns are contaminated by noise data, as observed in the previous method. These steps are explained in detail below.

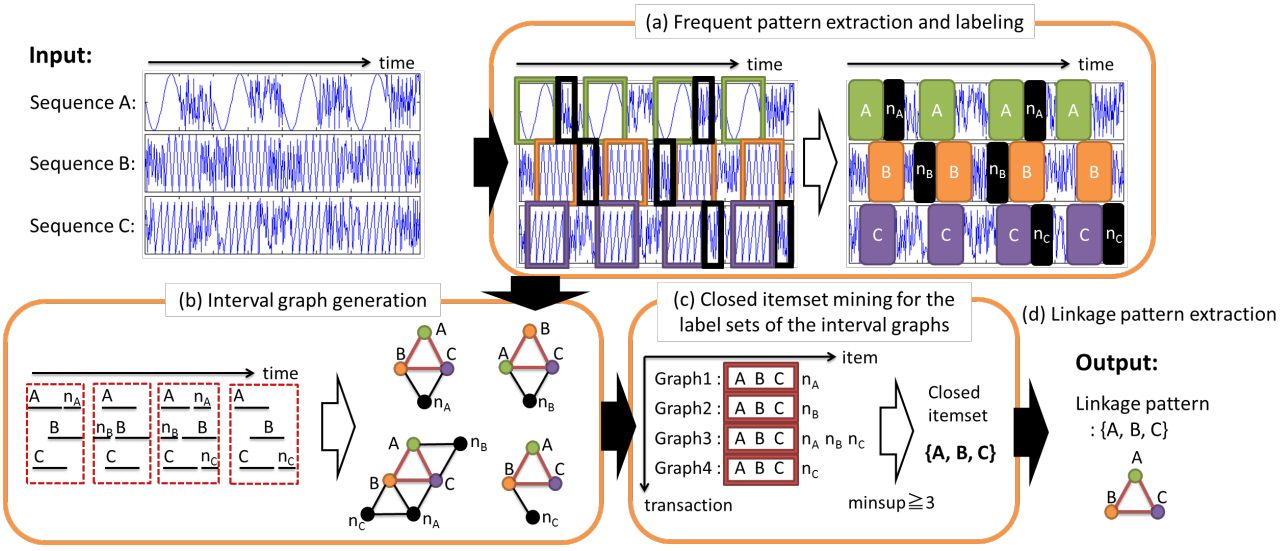


Figure. 2.1: Procedure of the proposed method

2.1 Preprocessing

First, normalization and discretization are executed on all sequential data in a preprocessing.

2.1.1 Normalization

In normalization, sequential data are converted to a scale from 0 to 1. Normalization is calculated as follows. In this equation, X_t means the value of time point. max and min mean values of maximum and minimum of sequence.

$$Normalize(X_t) = \frac{X_t - min}{max - min} \tag{2.1}$$

2.1.2 Discretization

In the discretization, the range of normalized data (0–1) is divided at the D stages, and a discrete value from 0 to $D - 1$ is allocated to each data. In this study, D was set to 50 fixed value.

2.2 Frequent Pattern Mining

2.2.1 Mannila's Algorithm

Next, repeatedly occurring frequent patterns are extracted from the sequential data using Mannila ' s algorithm [13]. This algorithm uses a window width w and a minimum number of occurrences θ as input parameters, where w and θ are natural numbers ≥ 2. Window width w is the length of the slice used to scan sequential data. The minimum number of occurrences θ is the minimum number of frequent patterns to be extracted. Mannila ' s algorithm finds frequent patterns that satisfy θ for a specified w.

2.2.2 Labeling

The labeling process applies the same label to the same frequent pattern. This process is performed after excluding frequent patterns with length less than w/2. When multiple frequent patterns occur within the same periods in the same sequential data, labeling is performed for the maximum length frequent pattern.

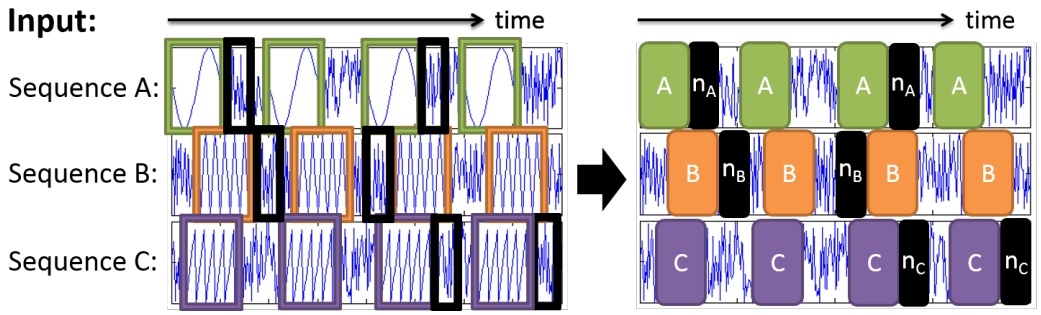
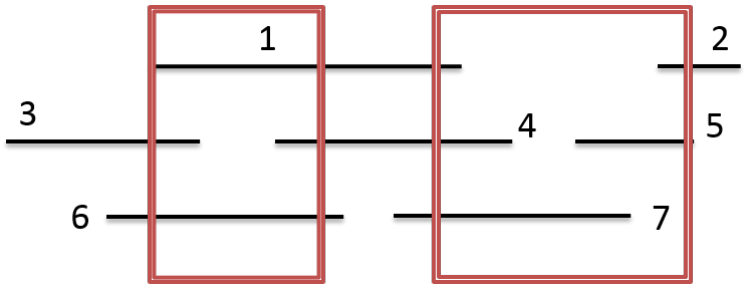


Figure. 2.2: Frequent pattern extraction and labeling

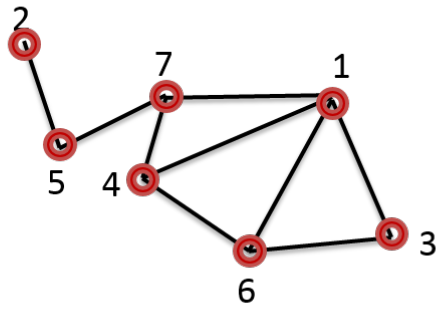
2.3 Interval Graph Generation

2.3.1 Definition of Interval Graph

In linkage pattern mining, we use the concept of interval graph. Figure.2.3 shows an example of interval graph. Interval graph is a subclass of chordal graph and is graph for representing interval overlaps. A node indicates an interval, and an edge means an overlap between two intervals. Figure.2.4 shows a step of interval graph generation. Here, a labeled frequent pattern is referred to as a label. In this step, interval graphs are generated from the interval representation of each label. An interval graph is obtained by associating each label with a node and an overlap of any two labels on the time axis between sequential data with an edge [18–20]. In other words, an interval graph is a set of frequent patterns that occur in a linked manner in the same period between different sequential data. The previous method outputs the interval graph with the highest frequency as a linkage pattern. However, frequent patterns that are accidentally constructed by noise (pseudo patterns) cause the following problems. If different pseudo pattern labels are attached to the same interval graphs, these interval graphs are considered completely different despite having an identical linkage pattern. This reduces the accuracy of linkage pattern mining.



(a) Interval representation



(b) Graph representation

Figure. 2.3: Interval graph representation

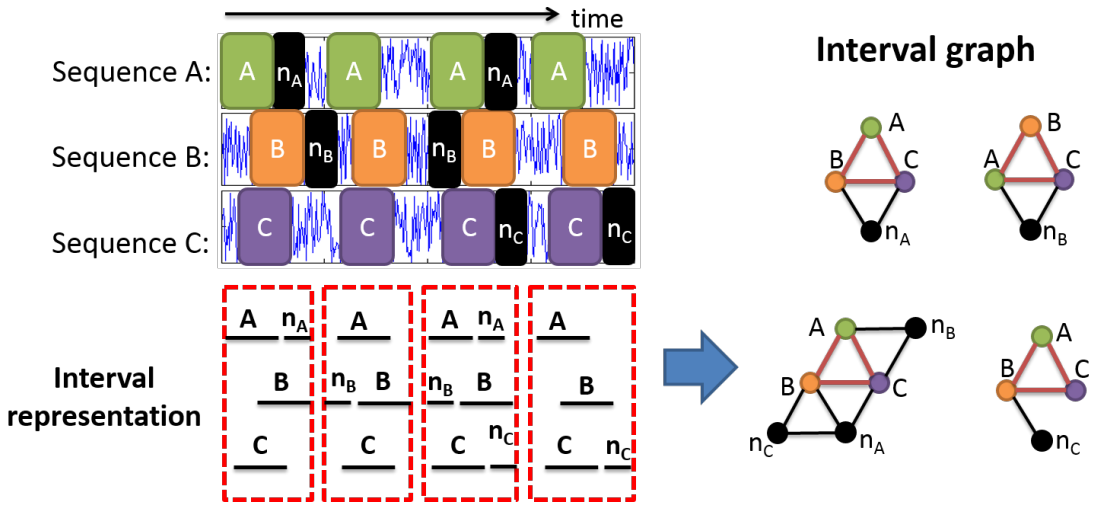


Figure. 2.4: Interval graph generation

2.4 Closed Itemset Mining

2.4.1 Definition of Closed Itemset

Let $I = \{1, 2, \cdots, n\}$ be a set of items. A transaction database on I is a set $T = \{t_1, t_2, ..., t_m\}$ such that each t_i is included in I. Each ti is called a transaction. A set $P \subseteq I$is called an itemset. A transaction including P is called an occurrence of P. The set of occurrences of P is expressed as $T(P)$. The size of a set of occurrences for P is referred to as the frequency of P.

An itemset P is called a closed itemset if no other itemset Q satisfies $T(P) = T(Q)$, $P \subseteq Q$. For a given minimum support constant (hereafter *minsup*), P is frequent if $|T(P)| \geq minsup$. A frequent and closed itemset is referred to as a frequent closed itemset.

2.4.2 Closed Itemset Extraction

Pseudo patterns tend to occur randomly on the time axis; thus, the probability that the same pseudo pattern will be included in multiple equivalent interval graphs is extremely low. Therefore, it is expected that pseudo patterns can be excluded by extracting label sets that occur commonly in multiple interval graphs. The proposed method extracts clear linkage pattern without the pseudo patterns by closed itemset mining on the obtained interval graphs.

Figure.2.5c shows the process of excluding pseudo patterns from interval graphs. Each interval graph is considered a transaction, and each node in the interval graph is considered an item. By applying closed itemset mining to this transaction database, we can extract the maximal node sets (closed itemsets) that are shared in minsup or more interval graphs. Finally, the closed itemset with the highest frequency is output as the linkage pattern. Thus, it is possible to extract linkage patterns with greater accuracy as randomly constructed pseudo patterns can be excluded. Figure.2.5c illustrates an example of how pseudo patterns nA, nB, and nC are excluded; only the authentic linkage patterns {A, B, C} are extracted.

In this study, we use the fast and exhaustive linear closed itemset miner (LCM) algorithm [21].

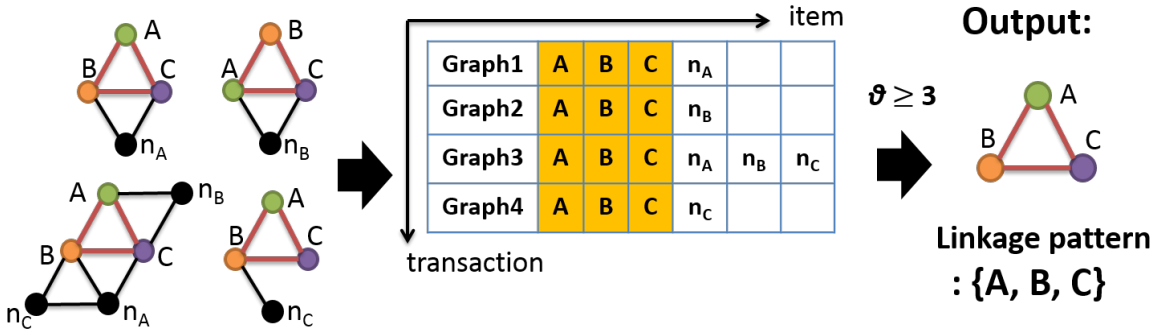


Figure. 2.5: Cloased itemset extraction

2.5 Experiment

2.5.1 Evaluation of Linkage Patten Mining

The proposed method was evaluated for extraction accuracy and computational time using artificially created sequential datasets.

Artificial Datasets

Each artificial dataset comprised three sequential data. The sequential data were generated by inserting 10 linkage patterns (embedded linkage patterns) into random sequential data created using uniform random numbers. For this experiment, we created five non-noise artificial datasets (Dataset1-Dataset5) that included no noise within the embedded linkage patterns. Figure.2.6 shows a section of each artificial dataset. The formats of linkage patterns embedded in each dataset are as follows. Dataset1 is an artificial dataset wherein equal length frequent patterns were embedded with the same start time across the three sequential data (Figure.2.6a). Dataset2 is an artificial dataset wherein equal length frequent patterns were embedded with different start times across the three sequential data (Figure.2.6b). Dataset3 is an artificial dataset wherein different length frequent patterns for each of the three sequential data were embedded at the same time (Figure.2.6c). Dataset4 is an artificial dataset wherein frequent patterns with different lengths for each of the three sequential data were embedded at different times (Figure.2.6d). Dataset5 is an artificial dataset wherein one or two types of frequent patterns were embedded with different lengths and different start times for each of the three sequential data (Figure.2.6e).

In addition, five artificial datasets (Dataset1_noiseDataset5_noise) that included noise in the embedded linkage patterns were created by adding fluctuations to each time point

in the linkage patterns. The fluctuations were generated using normal random numbers (SD = 0.01).

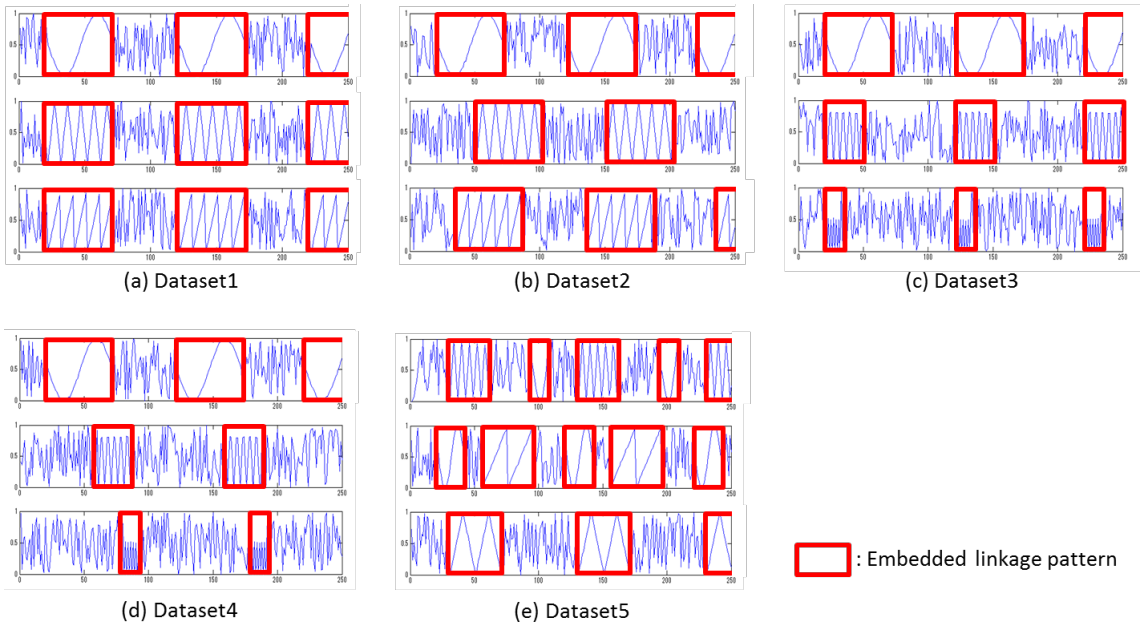


Figure. 2.6: Artificial datasets

Parameter Setting

For frequent pattern extraction, the minimum number of occurrences θ was fixed at 5, and the window widths w were set to natural numbers ≥ 3. For closed itemset mining, *minsup* were set to natural numbers ≥ 2.

Extraction Accuracy of Linkage Patterns

The extraction accuracies of the embedded linkage patterns for the previous and proposed methods were compared using the above 10 artificial datasets. *Precision*, *recall*, and *F-measure* were used as evaluation indexes. These indexes were calculated as follows.

$$\begin{aligned} Precision &= CDP/DDP \\ Recall &= CDP/EDP \\ F\text{-}measure &= 2 * Precision * Recall/Precision + Recall \end{aligned} \tag{2.2}$$

Here, CDP is the number of data points in the correctly detected areas of the embedded linkage patterns, DDP is the number of data points in the areas of the embedded linkage patterns detected by the method, and EDP is the number of data points in the embedded linkage patterns.

Evaluation of Computational Time

This experiment was conducted using the five noisy datasets (Dataset1_noise - Dataset5_noise).

The window width w significantly affected the computational time required to find frequent patterns [13]. First, we evaluated the computational time for the range of w described in Section 2.2.

In addition, sequential data length may also largely influence the computational time. Therefore, we increased the length by linking each dataset together and measured computational time when modifying up to 10,000 points in increments of 1,000.

2.5.2 Grid Search

Here, we describe the grid search method for the three parameters, i.e., w, θ, and *minsup*. The grid search is performed using five artificially created sequential noise datasets(Dataset1_noise - Dataset5_noise). The goal of this experiment was to find good parameter values that lead to high extraction accuracy.

Parameter Setting

For frequent pattern extraction, the minimum number of occurrences θ was set to natural numbers ≥ 5 in increments of 5, and the window width w values were set to natural numbers ≥ 3 in increments of 1. For closed itemset mining, *minsup* values were set to natural numbers ≥ 3 in increments of 2.

2.6 Results and Discussion

2.6.1 Evaluation of Linkage Patten Mining

Extraction Accuracy for Non-noise Datasets

Figures.2.7 and 2.8 are graphs of *precision*, *recall*, and *F-measure* in different w when the previous and proposed methods were applied to the five non-noise datasets. The *minsup* was fixed at 5. In these graphs, the results in the range $3 \leq w \leq 9$ are shown because no frequent patterns were extracted in $w \geq 10$. As we can see, the previous method shows unstable scores for different w values. This is caused by the pseudo patterns randomly formed by noise added to the embedded linkage patterns. In contrast, the proposed method demonstrates 100% extraction accuracy for $w > 4$. This means that the noises included in the interval graphs were suitably excluded by closed itemset mining.

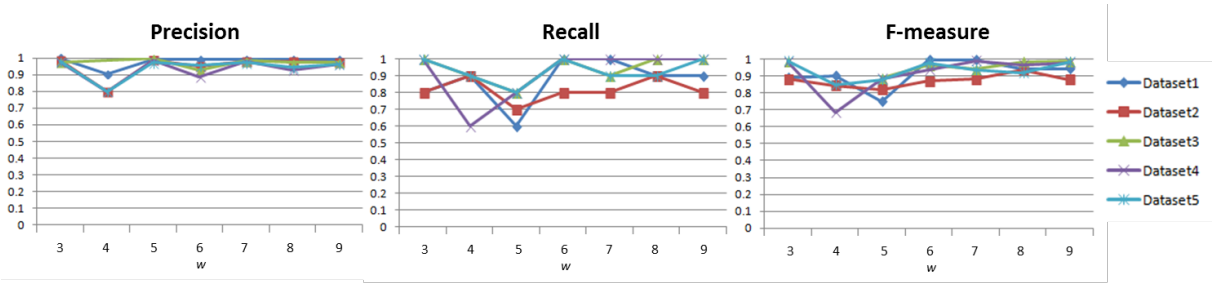


Figure. 2.7: Extraction accuracies in different w for the datasets without noise by the previous method

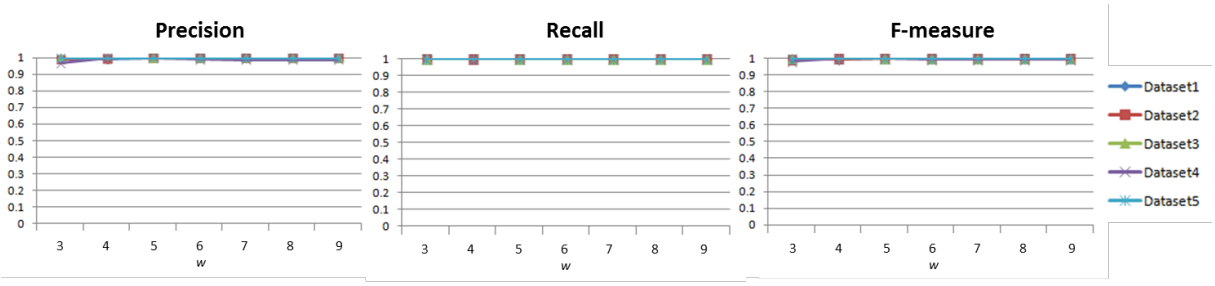


Figure. 2.8: Extraction accuracies in different w for the datasets without noise by the proposed method

Extraction Accuracy for Noise Datasets

This experiment was conducted using parameters ($minsup = 5$, $3 \leq w \leq 9$) same as those in the previous section. In the previous method, the accuracy of extracting linkage patterns was 0% for all datasets because only one interval graph was generated. This is because pseudo patterns exist throughout the sequential data. Thus, only the results of the proposed method are shown in this section. Figure.2.9 shows graphs of *precision*, *recall*, and *F-measure* in different w for the five datasets with noise (Dataset1_noise - Dataset5_noise). The *precision* values for all datasets are $\geq 80\%$ for all w values. In particular, when $w \geq 5$, the embedded linkage patterns are effectively extracted from all datasets because pseudo patterns are suitably excluded by closed itemset mining. Note that *recall* tends to decrease as w increases. In particular, when $w \geq 5$, *recall* decreases drastically for all datasets because the number of frequent patterns extracted from each sequence decreases drastically. Therefore, the obtained interval graphs are also reduced drastically. Note that *F-measure* decreases significantly with the drastic decline of *recall* values.

In addition, we investigated the impact of *minsup* on the extraction accuracy. Figure.2.10 shows graphs of *precision*, *recall*, and *F-measure* in different *minsup*s. In this experiment, the w was fixed at 5. The *precision* tends to increase with increasing *minsup* in all the datasets. In contrast, the *recall* decreases dramatically with increasing *minsup*. In particular, a rapid decrease in the scores is observed in $minsup \geq 5$. The *F-measure* is a similar tendency to the *recall* and especially shows high scores in the range of $2 \leq minsup \leq 4$.

From the above results, we can see that w and *minsup* should be fixed at a smaller value to obtain higher extraction accuracy.

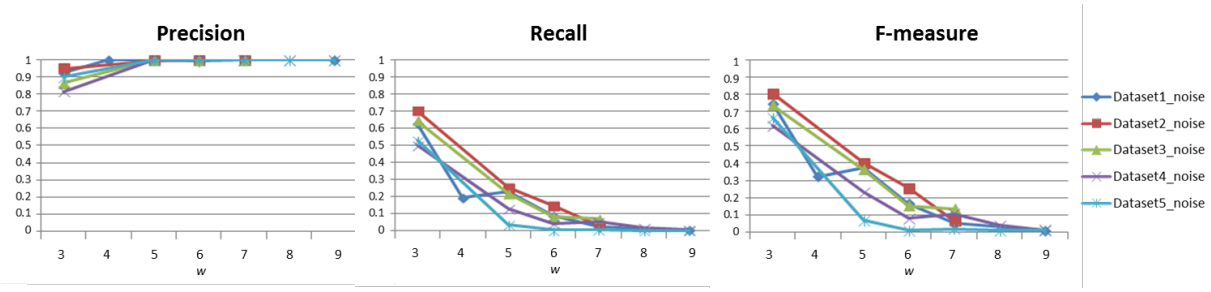


Figure. 2.9: Extraction accuracies in different w for the datasets with noise by the proposed method

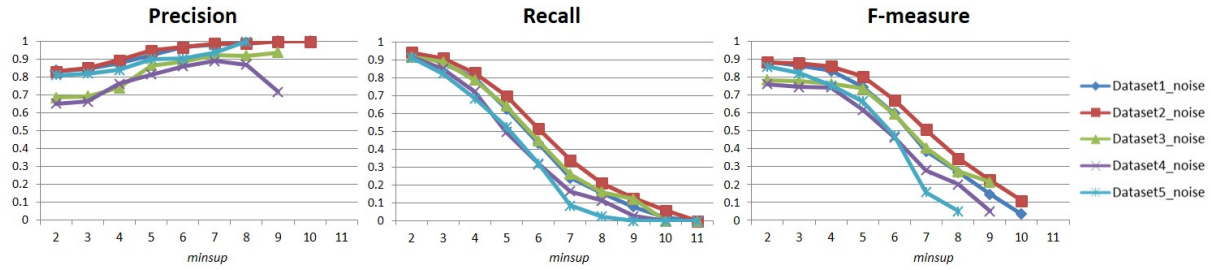


Figure. 2.10: Extraction accuracies in different minsup for the datasets with noise by the proposed method

Impact of Window Width on Computational Time

Figure.2.11 shows graphs of computational times when w was varied. In this experiment, $minsup$ was set to 5. In addition to the total computational time required for all steps in the proposed method, these graphs show the computational time for Steps (a), (b), and (c). Note that Step (a) is frequent pattern extraction and labeling, Step (b) is interval graph generation, and Step (c) is linkage pattern extraction based on closed itemset mining. As we can see, the total computational time is strongly affected by the computational time of Step (a) and increases drastically with increasing w because Step (a) must check labels in a combinatorial manner to find frequent patterns. On the other hand, the computational times of Steps (b) and (c) are considerably shorter than Step (a) and relatively stable against the increased w. This is due to the following reasons. First, Step (c) only detects the overlapped intervals along the time axis and therefore can be executed in linear time for the sequential data length. Furthermore, with regard to Step (c), besides the closed itemset enumeration algorithm LCM being exceptionally fast, the size of the transaction database for interval graphs was small (only tens to hundreds of transactions). Thus, computational time is highly dependent on the time required to extract frequent patterns; however, it is possible to execute within a realistic time by reducing the w value.

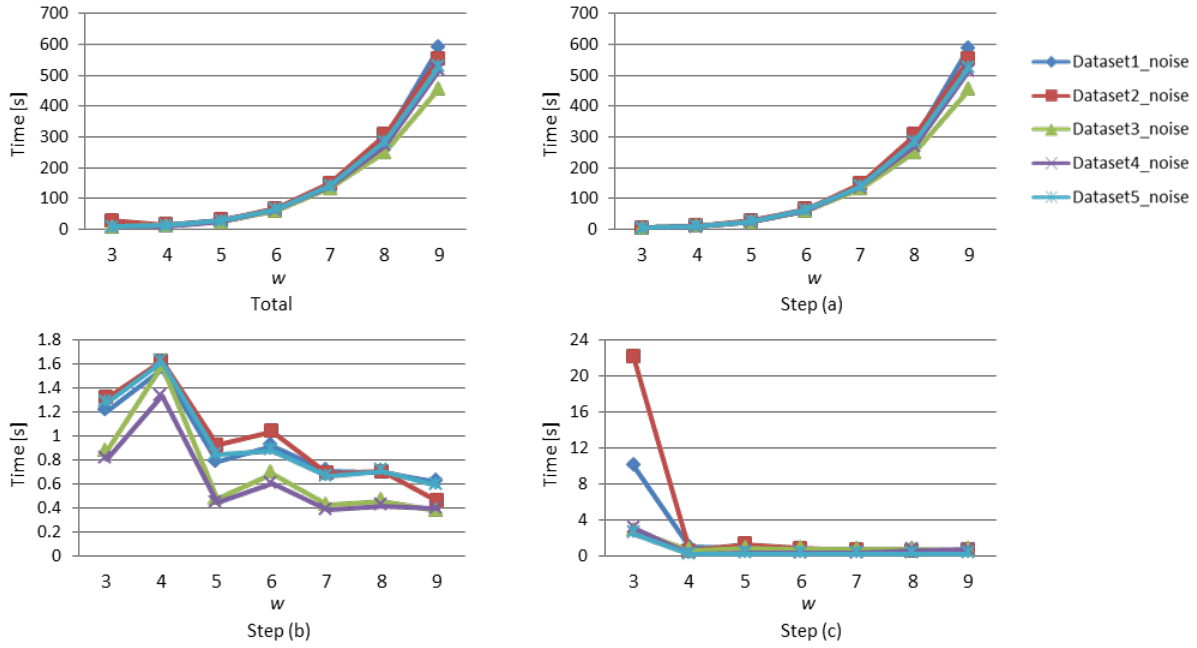


Figure. 2.11: Computational times in different w

Impact of Sequential Data Length on Computational Time

Figure.2.12 shows graphs of computational time for each step, including the total time when sequential data length was changed. In this experiment, w and *minsup* were set to 5. As we can see, Steps (a) and (b) increase linearly with increased sequential data length. However, Step (a) requires more computational time than Step (b) owing to the combinatorial search in frequent pattern extraction. For Step (c), the computational times are considerably less than Steps (a) and (b) although there are major fluctuations related to sequential data length. This is because the size of the transaction database changed depending on the number of extracted interval graphs. From the above, we can see that the computational time of the proposed method increases linearly with increased sequential data length. However, the computational time required to extract frequent patterns constitutes a large proportion of the proposed method ' s total computational time. Increasing the speed of the frequent pattern mining algorithm will certainly become an issue when applying this method to large-scale real data.

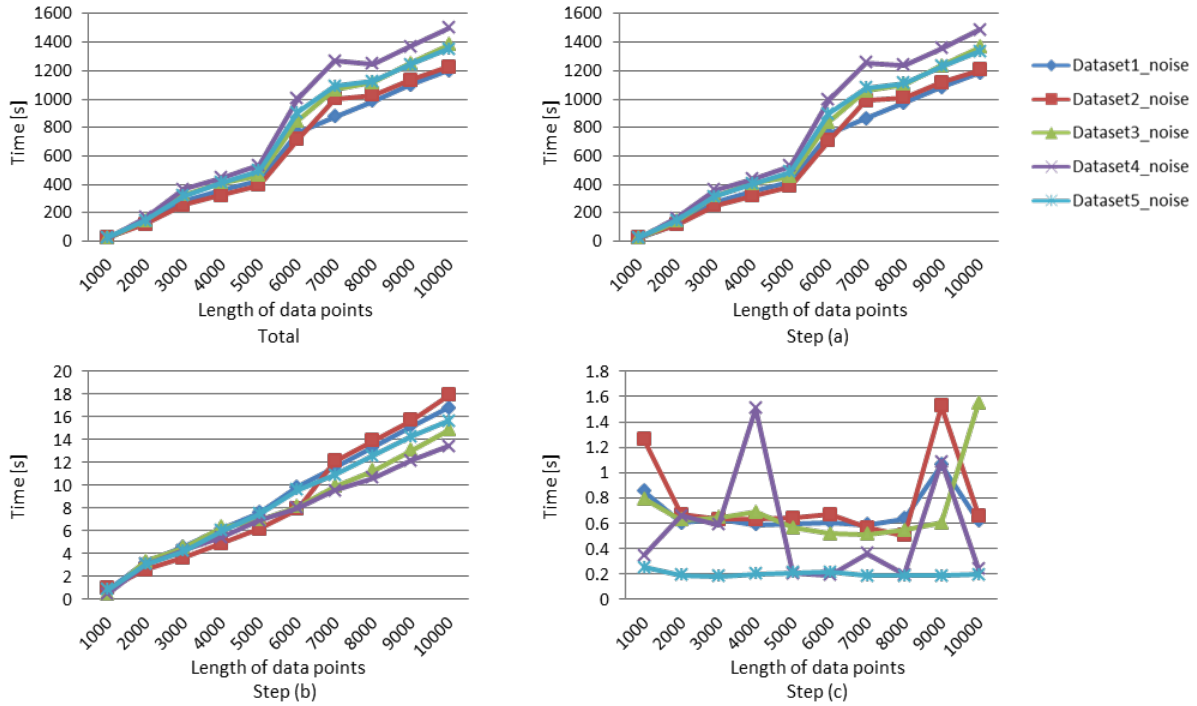


Figure. 2.12: Computational times in sequential data of different lengths

2.6.2 Grid Search

Figures.2.13, 2.14, and 2.15 show the grid search results of *precision*, *recall*, and *F-measure* for the five datasets(Dataset1_noise - Dataset5_noise), respectively. The two-dimensional graphs in these figures are the scores for different combinations of the two parameters, w and θ, and the respective *minsup* values are shown. In these figures, for $w > 9$ and $\theta > 30$, the results are not shown because it required too much computational time (several hours or days) for each combination of parameters. In addition, the results for $minsup > 11$ are not shown because no frequent pattern was extracted.

As described in Section 2.4, *minsup* is the minimum value (a threshold) of the number of interval graphs (transactions) having a set of common labels (nodes). Figures.2.13, 2.14, and 2.15 show that the extraction accuracy with large *minsup* indicates unstable and low scores for all the indexes. Large *minsup* results in linkage patterns composed of a small set of nodes, i.e., it increases the possibility of extracting false linkage patterns. This causes decreased *precision*. Furthermore, large *minsup* also decreases *recall* because only high-frequent linkage patterns are targeted. In contrast, extraction accuracy with small *minsup* shows higher score regardless of the combinations of w and θ. In many cases, small *minsup* yields false or noise-contaminated linkage patterns. However, the proposed

method can remove such pseudo patterns adequately using the closed itemset mining process and enables high and stable accuracy. From the above, we discuss only the results for small *minsup* (i.e., 3, 4, and 5).

From the results of *precision*, we consider that w should be set to smaller values to obtain high *precision*. This is based on the fact that large w does not show clear superiority compared to small w, in addition to requiring significant computational time. For θ, extremely small or large values should be avoided due to unstable performance. θ values that are too small yield many false linkage patterns composed of low-frequency patterns, and overly large θ makes it difficult to detect frequent patterns composing true linkage patterns. Thus, we consider that an adequate θ value should be selected by trial and error in order from smaller to larger values.

Recall shows relatively clear results compared to that of *precision*. It shows high scores when both w and θ are small. This means that the frequent pattern mining process focuses on searching for short length and low-frequency patterns. In other words, we can obtain many possible frequent patterns that can be components of a linkage pattern, i.e., *recall* increases. However, when w and θ are large values, *recall* decreases because the number of extracted frequent patterns decreases. Thus, both w and θ should be set to smaller values to obtain higher *recall*.

F-measure is the harmonic mean of *precision* and *recall*. From these graphs, we can see that smaller w and θ should be used to obtain high extraction accuracy.

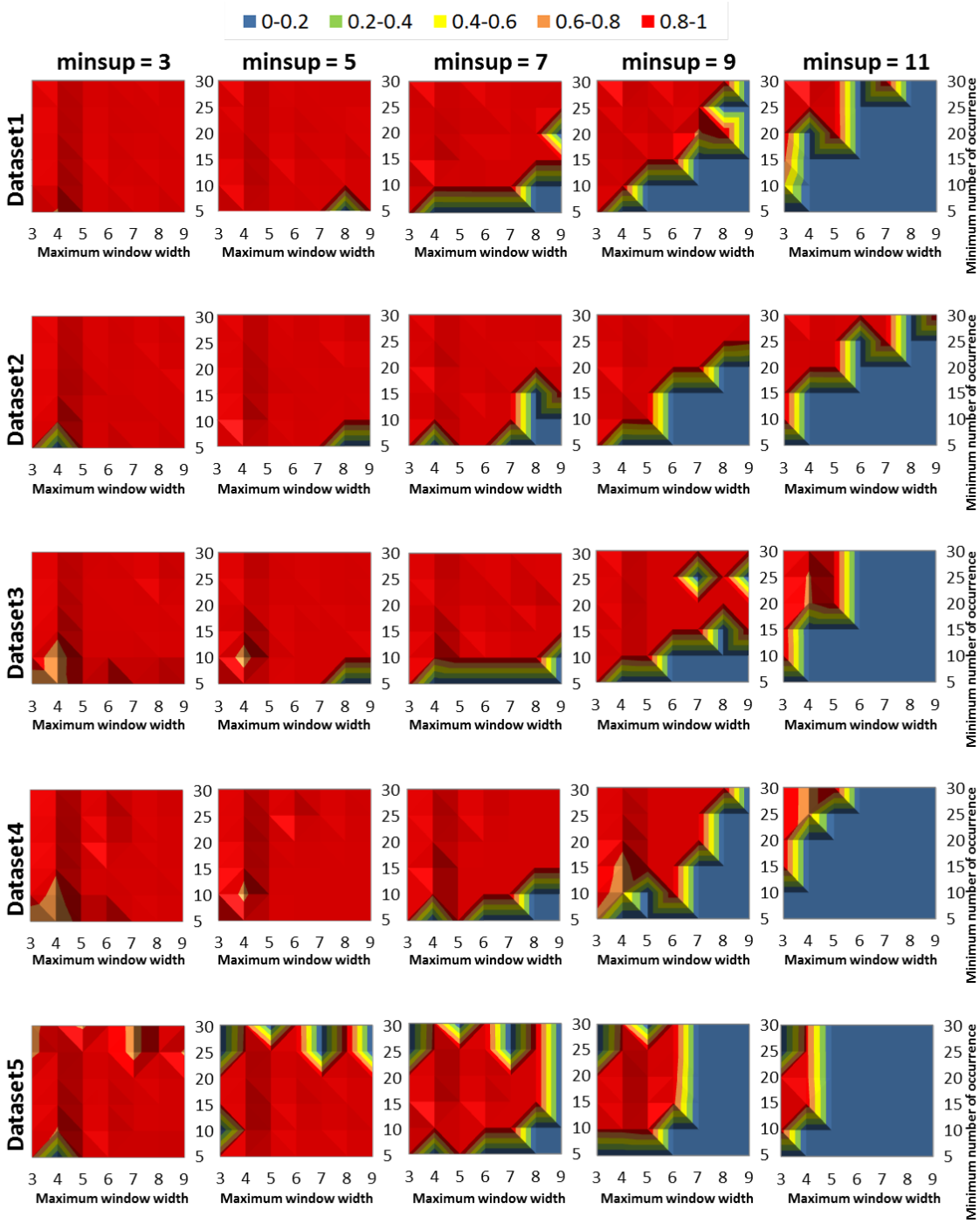


Figure. 2.13: Grid search results for *precision*

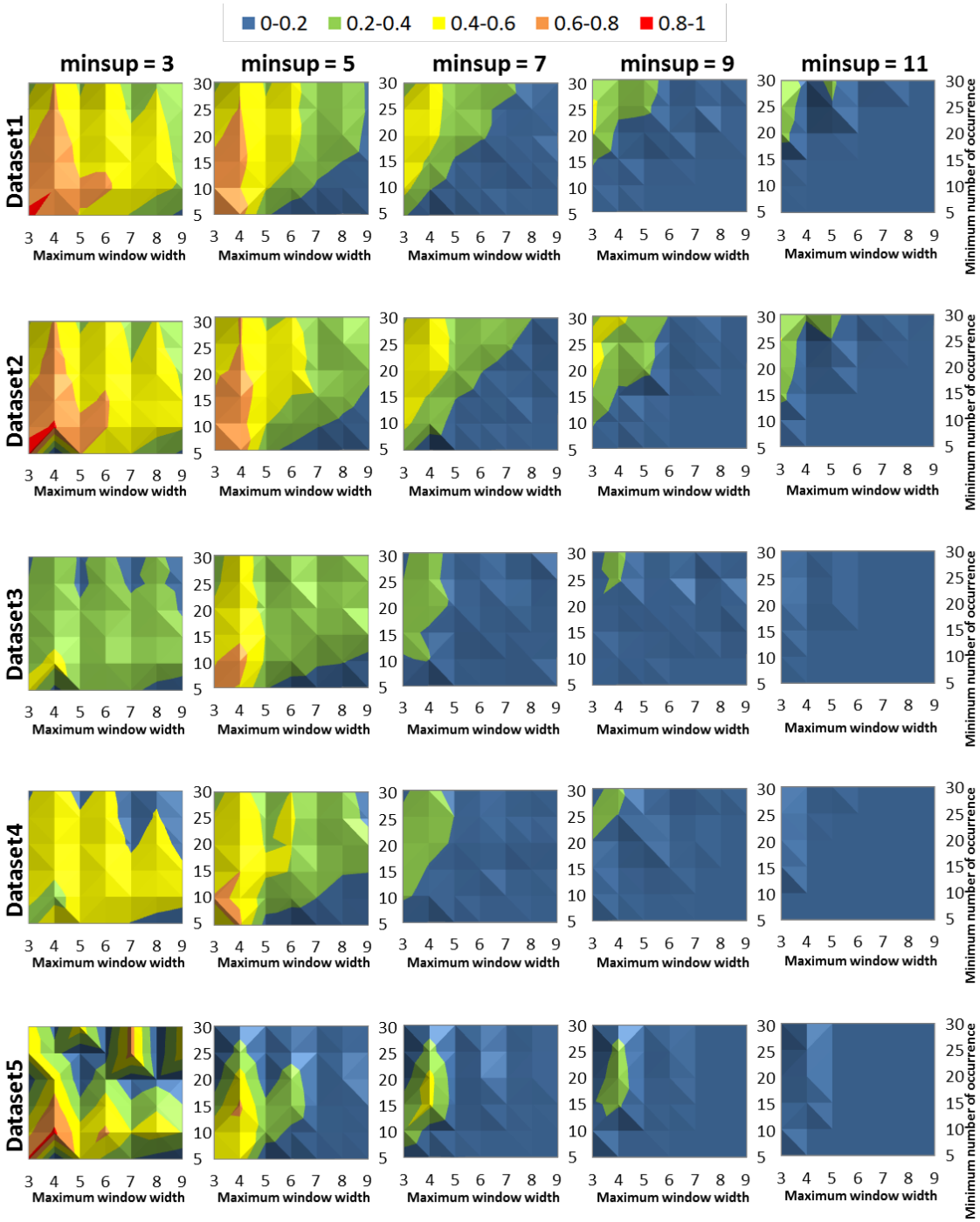


Figure. 2.14: Grid search results for *recall*

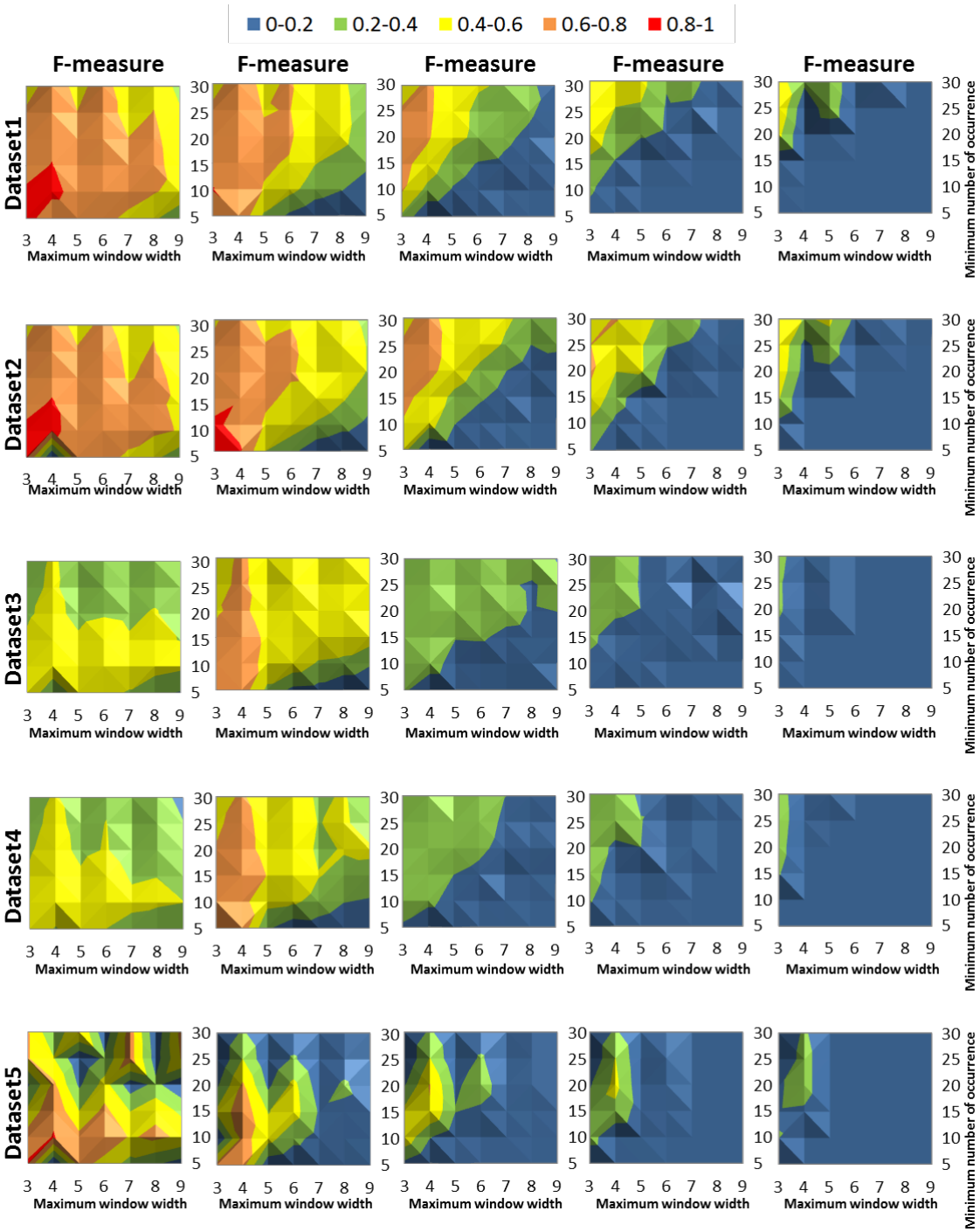


Figure. 2.15: Grid search results for *F-measure*

2.7 Conclusion

2.7.1 Evaluation of Linkage Pattern Mining

We proposed a new noise-robust linkage pattern mining method based on closed itemset mining. In the proposed method, closed itemset mining is employed to exclude pseudo patterns generated by noise/fluctuations and obtain only frequent and maximal patterns among different interval graphs. In our first experiment, we compared the performance of the previous and proposed methods using artificial datasets. As a result, it was shown that the proposed method can appropriately detect linkage patterns with noise that were not detected by the previous method. Furthermore, we found that w and *minsup* should be fixed at a smaller value to obtain higher extraction accuracy. In our second experiment, we measured computational time using five datasets with noise when the window width w and sequential data length were varied. As a result, we observed that computational time increases as w and sequential data length increase. Furthermore, in the proposed method, the impact of introducing closed itemset mining on computational time is substantially small.

In future, we will address increasing the speed of the frequent pattern mining algorithm. In addition, we will apply the method to large-scale real sequential data that includes noise/fluctuations, such as vital data and crustal movement data. The practical applicability of the proposed method will also be evaluated in terms of extraction accuracy and computational time

2.7.2 Grid Search

Our linkage pattern mining method requires three parameters, w, θ, and *minsup*. In our previous study [18], we used empirically-selected parameters. However, the extraction accuracy of linkage patterns is significantly affected by the combination of these parameters.

In this study, we conducted gird search experiments to investigate better combinations of the three parameters to provide high extraction accuracy. In the experiments, three indexes, i.e., *precision*, *recall*, and *F-measure*, were evaluated in different combinations of the parameters using five artificial sequential datasets. As a result, the following findings were obtained.

1) *minsup* should be set to a small value to obtain stable and high extraction accuracy. This is a common finding among the three indexes.

2) w should be set to a small value to obtain high precision. However, θ needs to be determined in order from the smallest value ($minsup = 3$).

3) Both w and θ should be set to small values to obtain high *recall*.

4) *F-measure* is the harmonic mean of *precision* and *recall*. Both w and θ should be set to small values to obtain high *F-measure*.

In the future, we will address increasing the speed of the frequent pattern mining algorithm and apply the method to large-scale real sequential data with noise/fluctuations, such as vital data and crustal movement data.

3 Application to ECG

3.1 ECG Data

ECG(Electrocardiography) is a tracing representing the heart's electrical action derived by amplification of the minutely small electrical impulses normally generated by the heart [29]. In other word, ECG is a recording of the electrical activity of the heart over a period of time using electrodes placed on the skin. ECG is composed of twelve sequences that are called electrodesleads. Electrodes-leads are measured from twelve specific sites of the body. Electrodes-leads consist of twelve sequences:standard limb leads-I, II, III, augmented limb leads- ${}_aV_r$, ${}_aV_l$, and ${}_aV_f$, and precordial or chest leads-V_1 to V_6. Typically, an ECG in each sequence consists of three major components: the P wave, which indicates atrial depolarization, the QRS complexventricular depolarization, and the T waveventricular repolarization [25]. Figure.3.1 shows the composition of ECG.

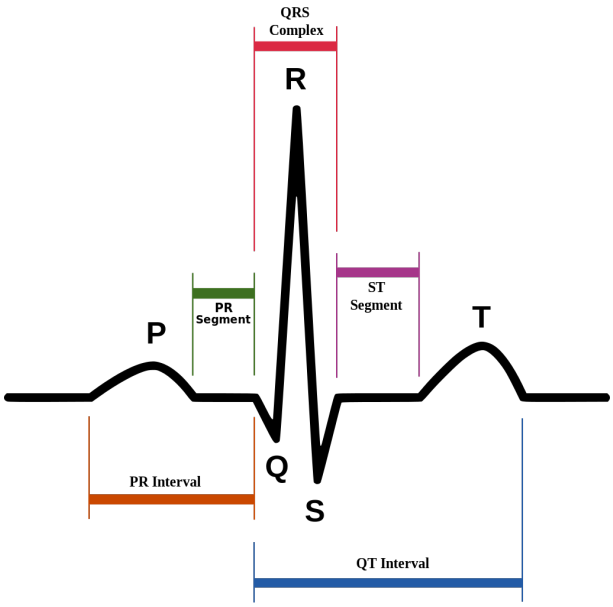


Figure. 3.1: Artificial ECG data

3.1.1 Healthy ECG Data

In this study, we use healthy ECG data for comparing with disease ECG data. Healthy data is composed of twelve sequences ($I, II, III, _aV_r, _aV_l, _aV_f$, and V_1 to V_6). The length of each sequence was set to 10000. This means that the number of data points is 10000. Figure.3.2 shows the twelve sequences of healthy ECG data. Figure.3.3 – 3.14 show the each sequence of healthy ECG data.

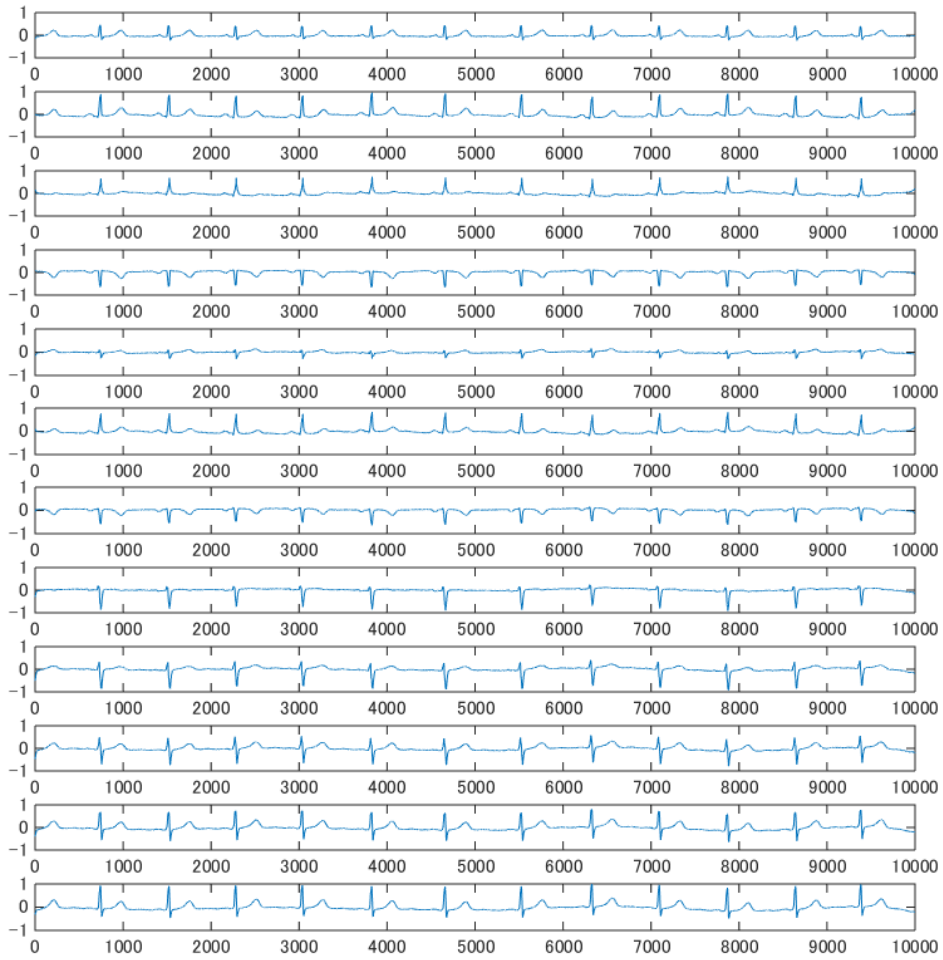


Figure. 3.2: Healthy data (Sequence 1 - 12)

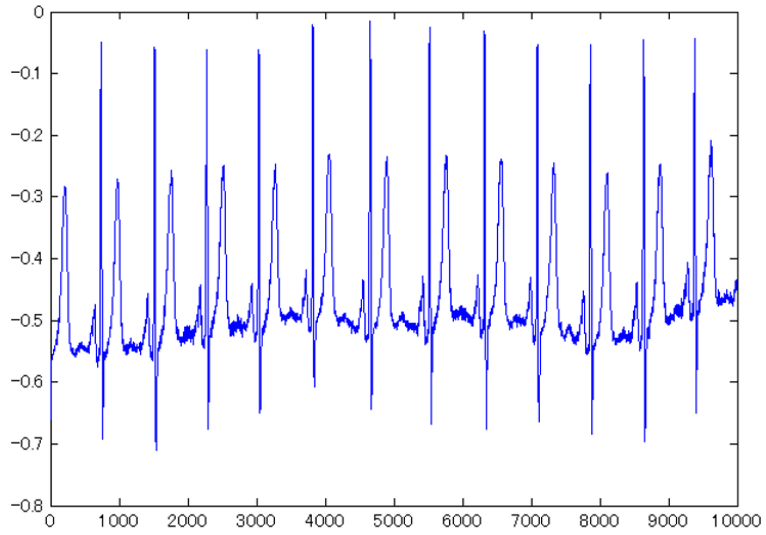


Figure. 3.3: Healthy data (Sequence 1)

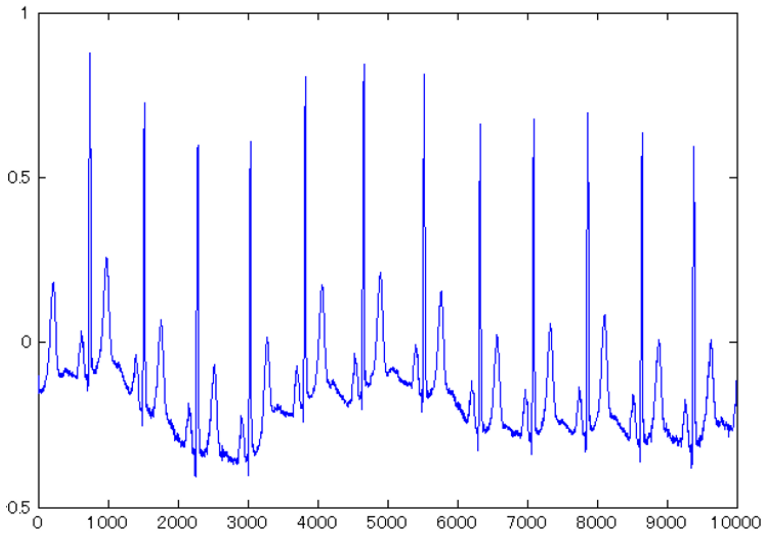


Figure. 3.4: Healthy data (Sequence 2)

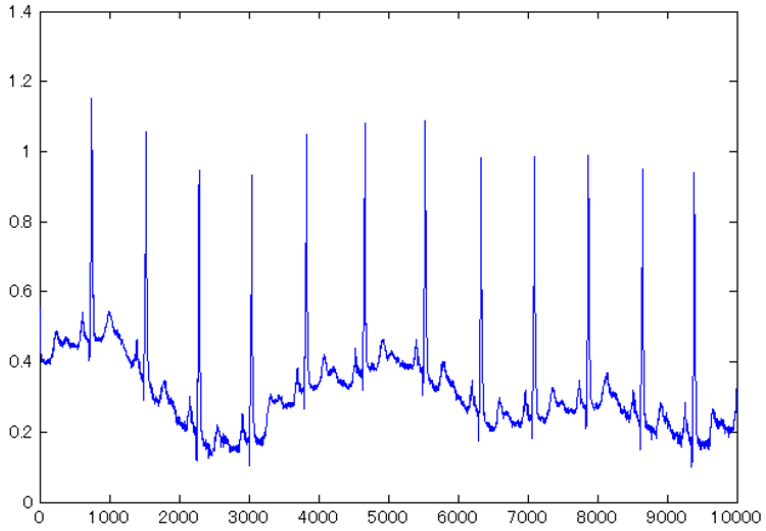


Figure. 3.5: Healthy data (Sequence 3)

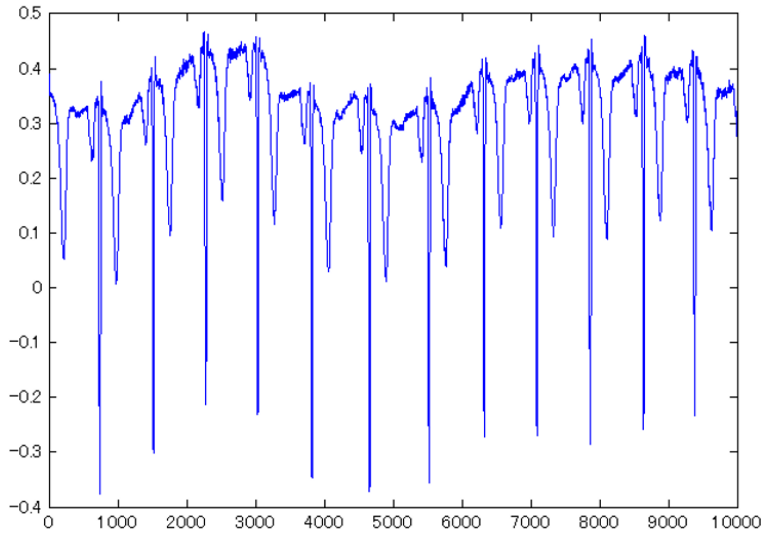


Figure. 3.6: Healthy data (Sequence 4)

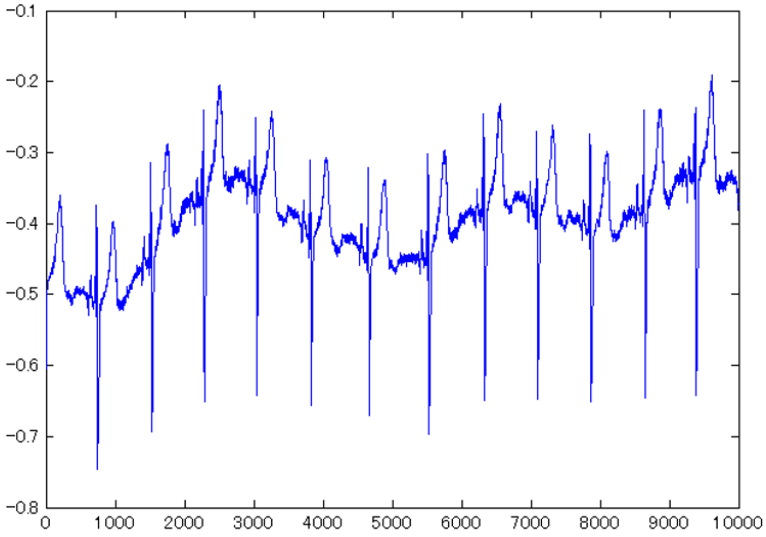


Figure. 3.7: Healthy data (Sequence 5)

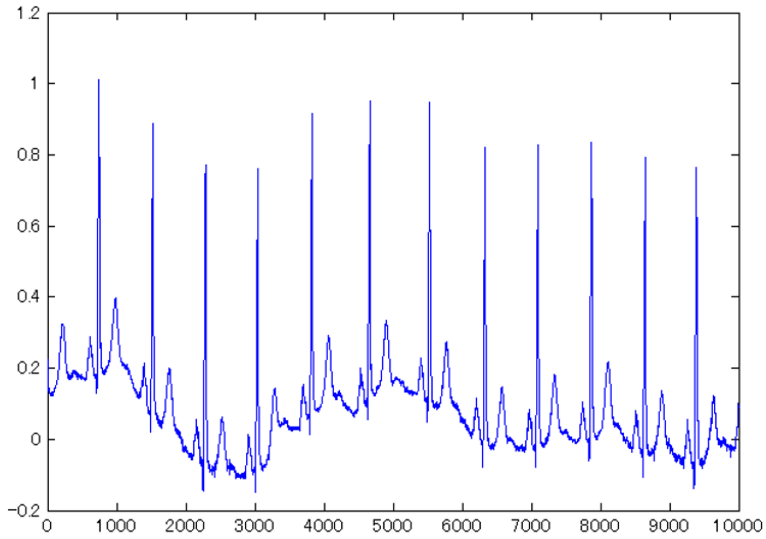


Figure. 3.8: Healthy data (Sequence 6)

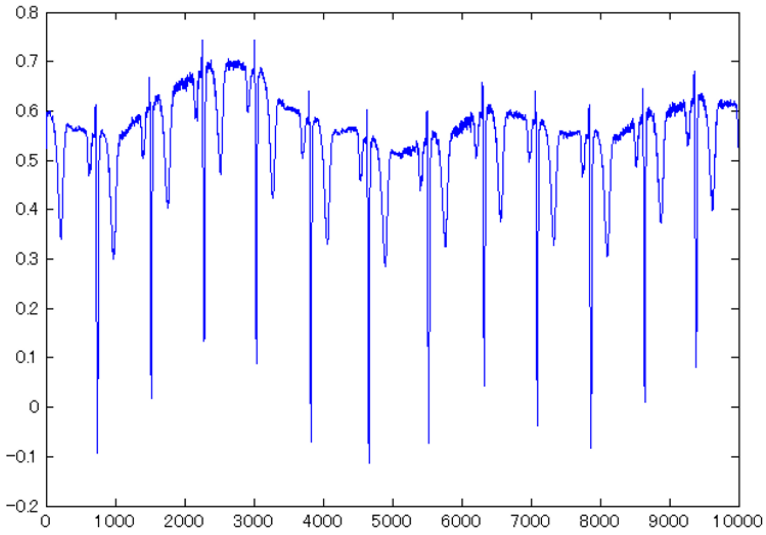


Figure. 3.9: Healthy data (Sequence 7)

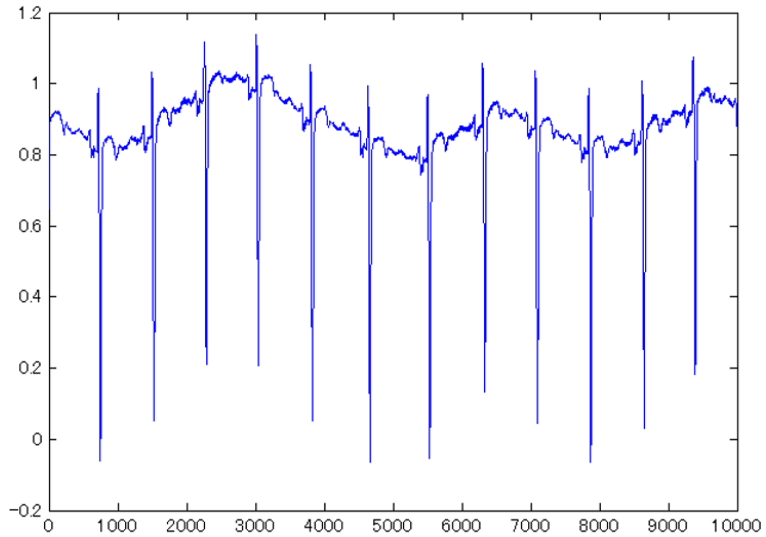


Figure. 3.10: Healthy data (Sequence 8)

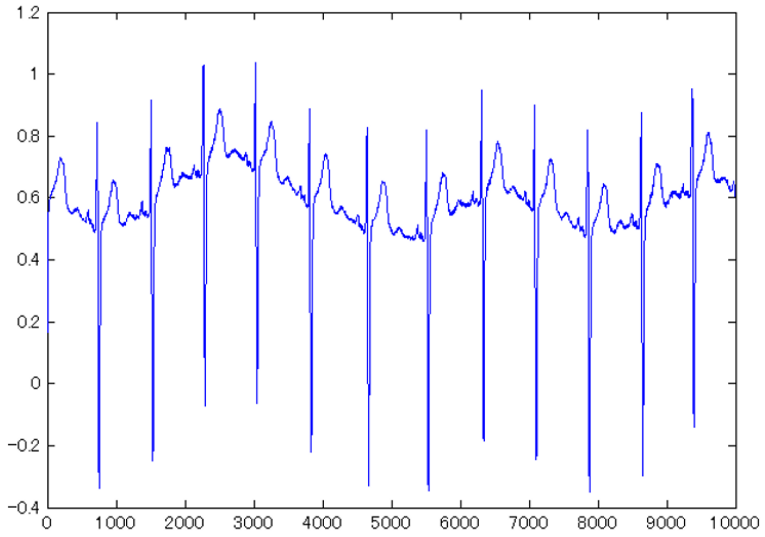


Figure. 3.11: Healthy data (Sequence 9)

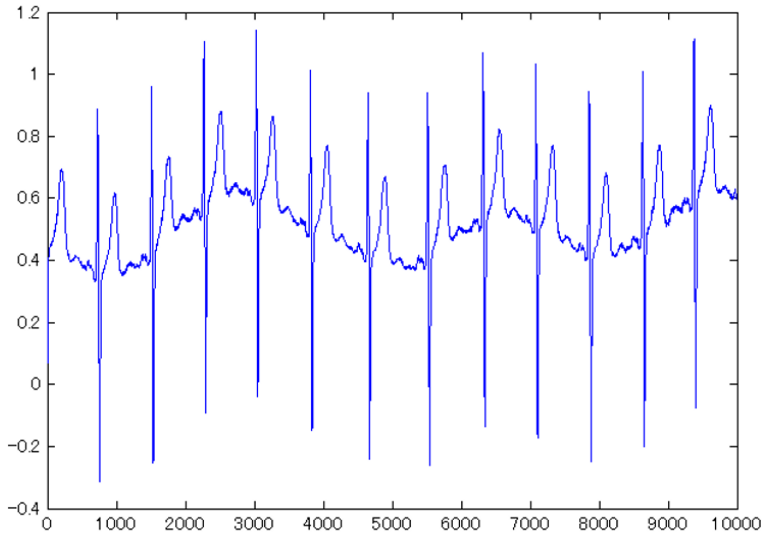


Figure. 3.12: Healthy data (Sequence 10)

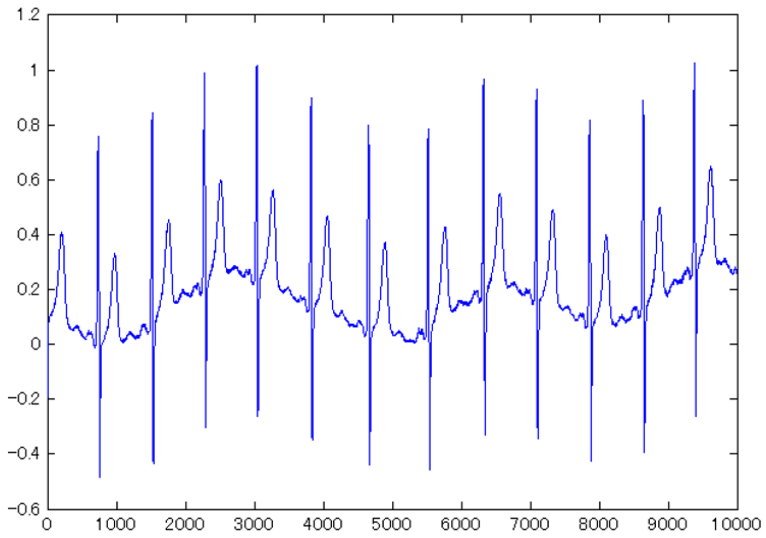


Figure. 3.13: Healthy data (Sequence 11)

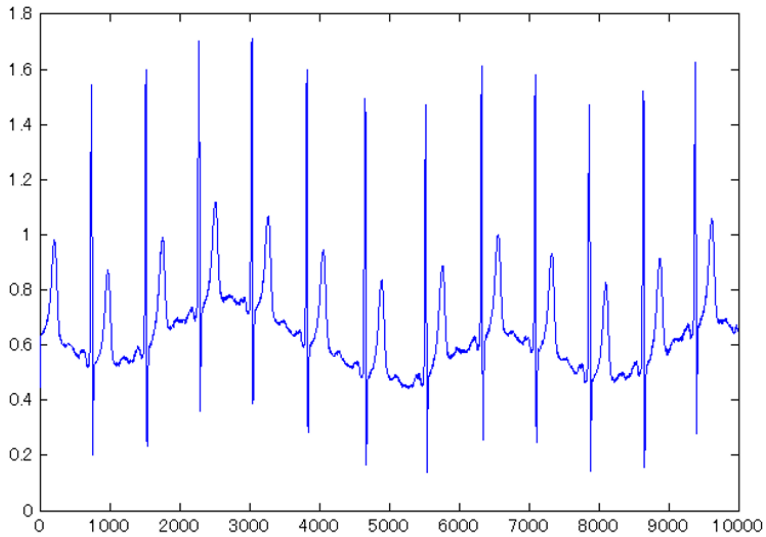


Figure. 3.14: Healthy data (Sequence 12)

3.1.2 Disease Data

In this study, a disease ECG data was used for detecting abnormal waveform using linkage pattern mining method. Information concerning patients are shown in detail below.

- age: 44
- sex: female
- ECG date: 13/05/1991
- Infarction date (acute): 12/05/1991
- Reason for admission: Myocardial infarction
- Acute infarction (localization): inferior
- Former infarction (localization): no
- Additional diagnoses: no
- Smoker: yes
- Number of coronary vessels involved: 1

As described in healthy ECG data, disease data is composed of twelve sequences (I, II, III, aVr, aVl, aVf, and V1 to V6). The length of each sequence was set to 10000. Figure.3.15 shows the healthy ECG data. Figure.3.16 – 3.27 show the each sequence of disease ECG data.

Preceding studies using interval between QRS wave and R wave by feature quantity [26–28]. However, some disease ECG indicating features between P wave and T wave. To detect features of heart disease, it is necessary to extract features these waves [23].

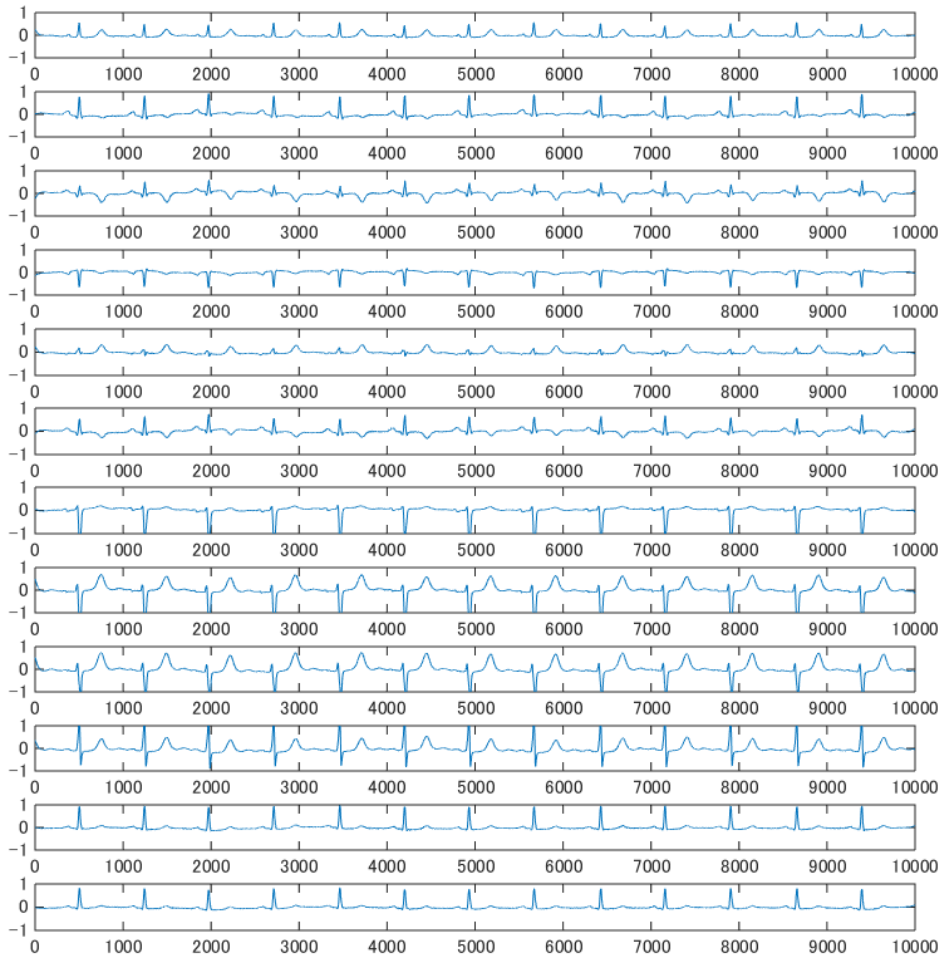


Figure. 3.15: Disease data (Sequence 1 - 12)

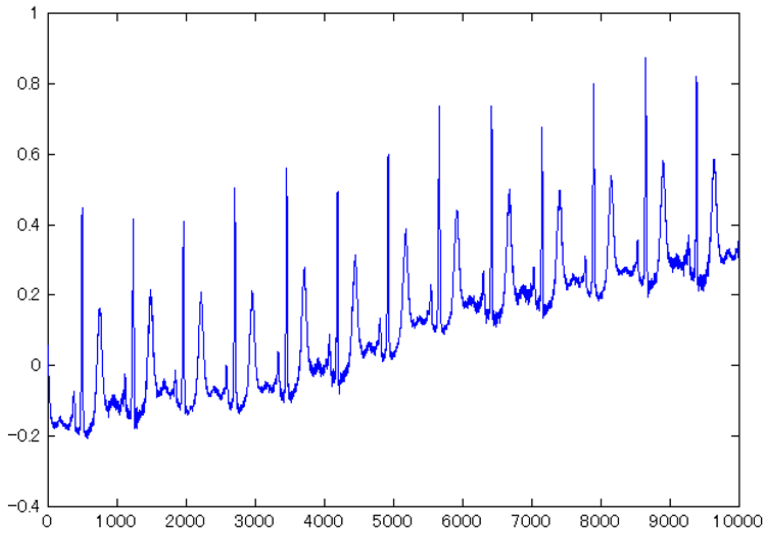


Figure. 3.16: Disease data (Sequence 1)

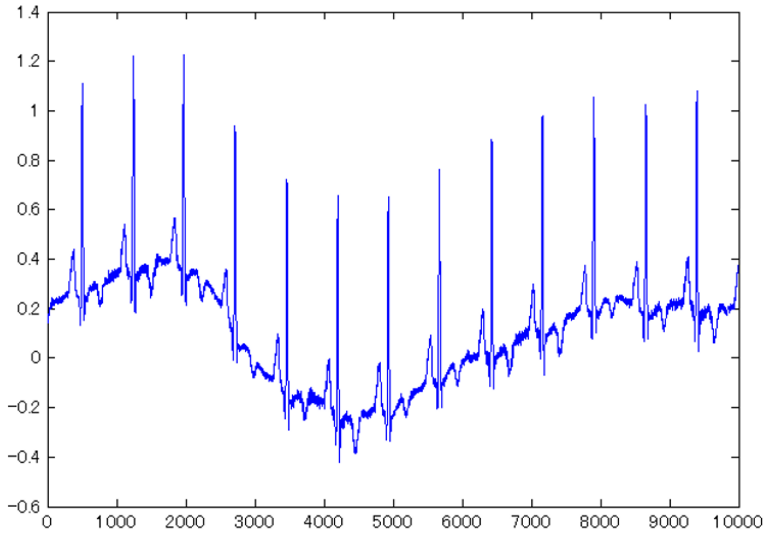


Figure. 3.17: Disease data (Sequence 2)

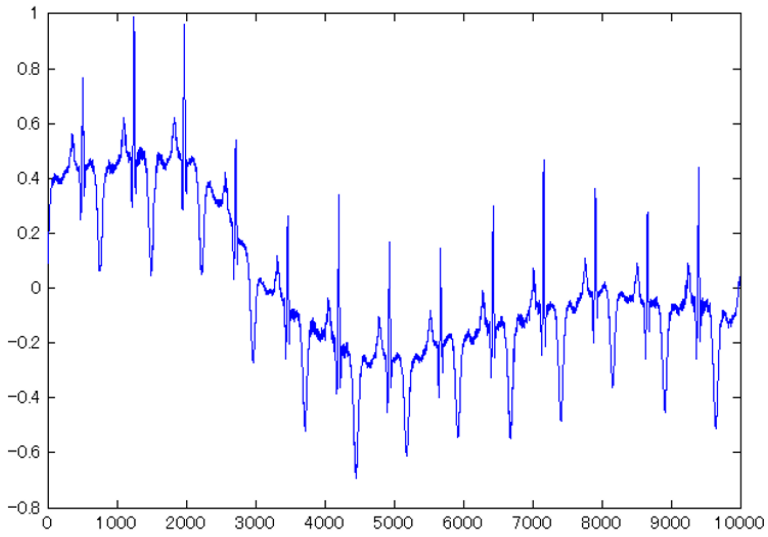


Figure. 3.18: Disease data (Sequence 3)

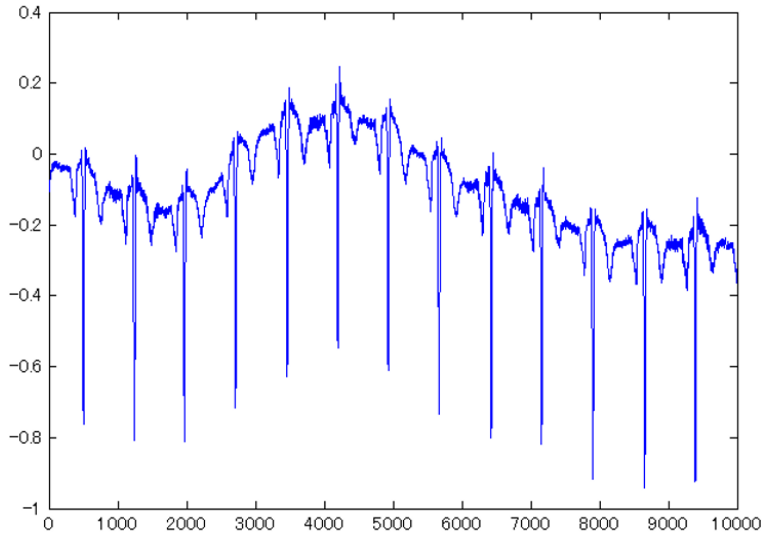


Figure. 3.19: Disease data (Sequence 4)

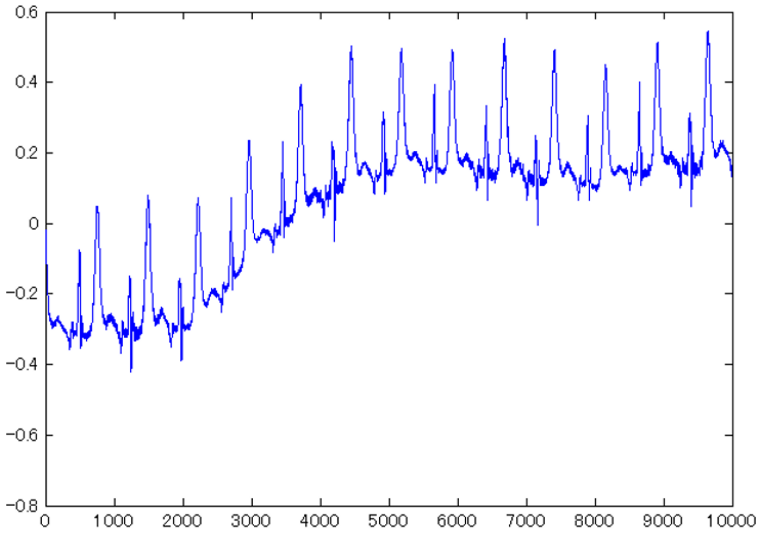


Figure. 3.20: Disease data (Sequence 5)

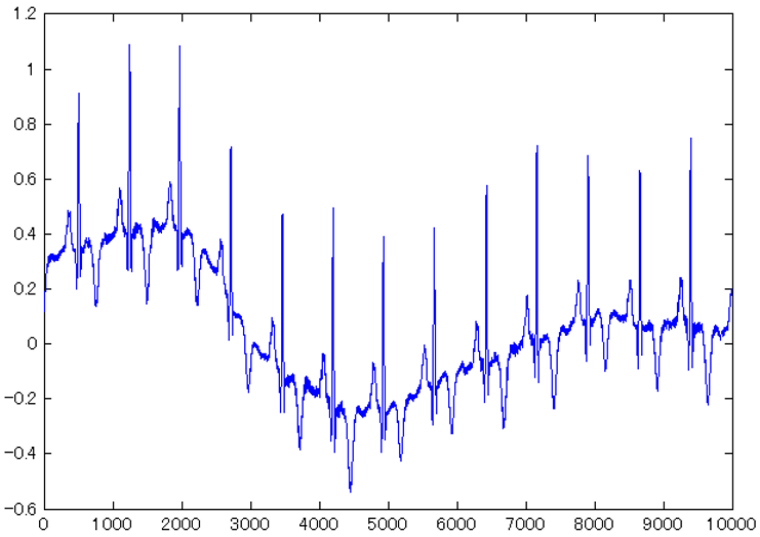


Figure. 3.21: Disease data (Sequence 6)

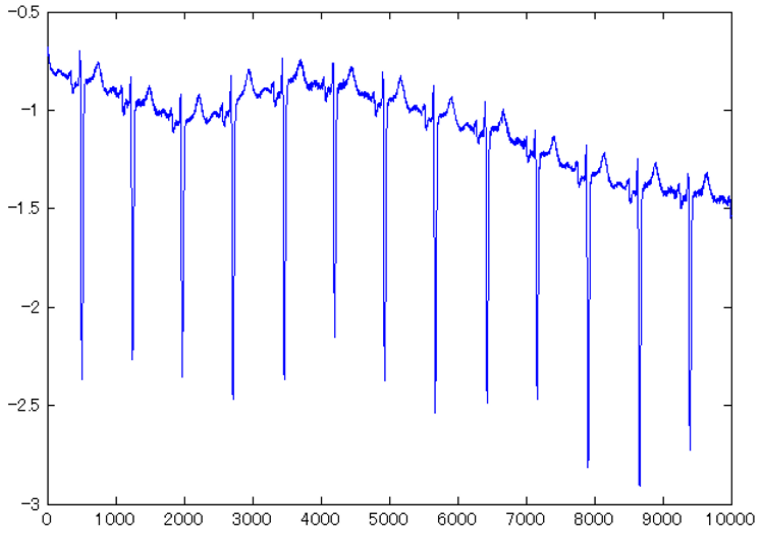


Figure. 3.22: Disease data (Sequence 7)

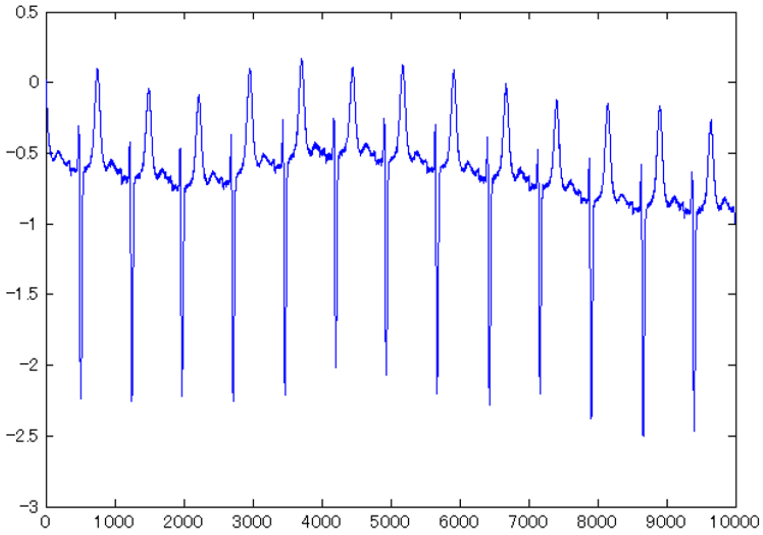


Figure. 3.23: Disease data (Sequence 8)

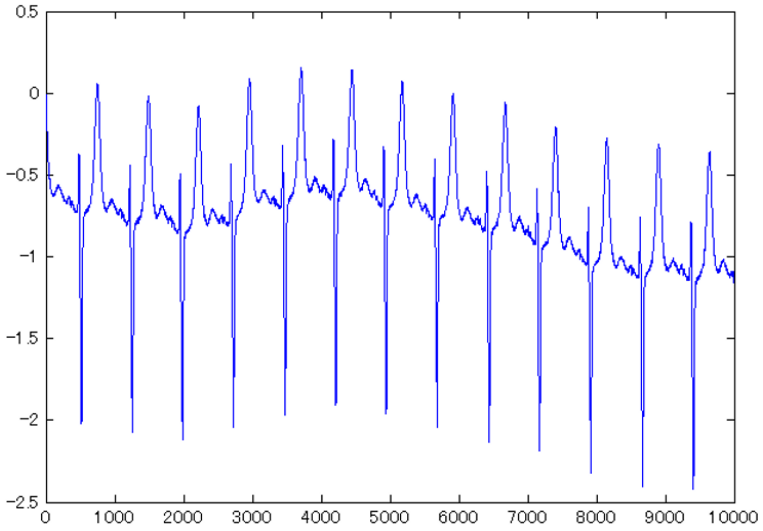


Figure. 3.24: Disease data (Sequence 9)

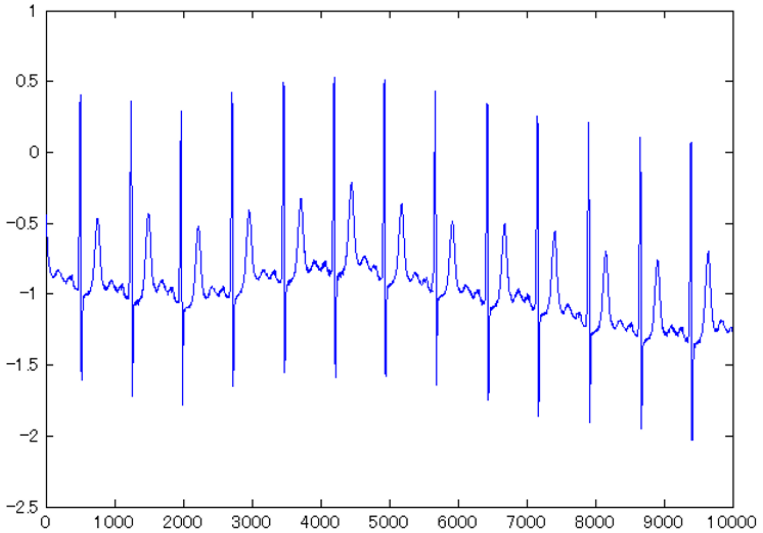


Figure. 3.25: Disease data (Sequence 10)

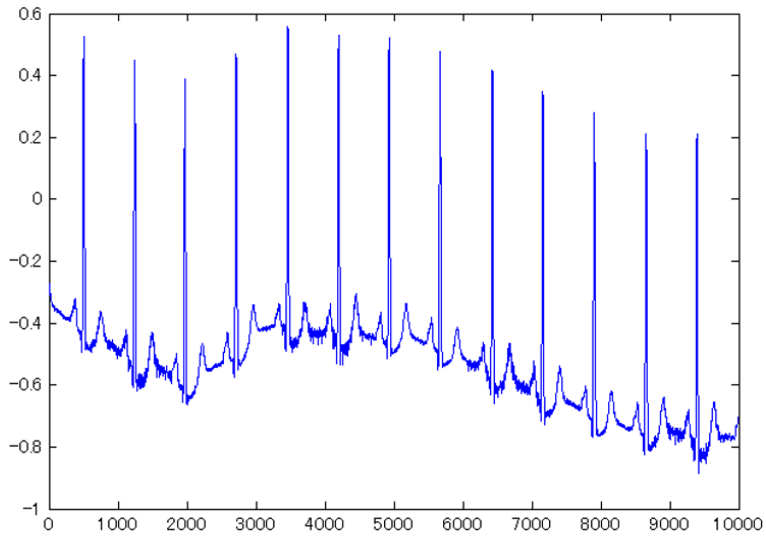


Figure. 3.26: Disease data (Sequence 11)

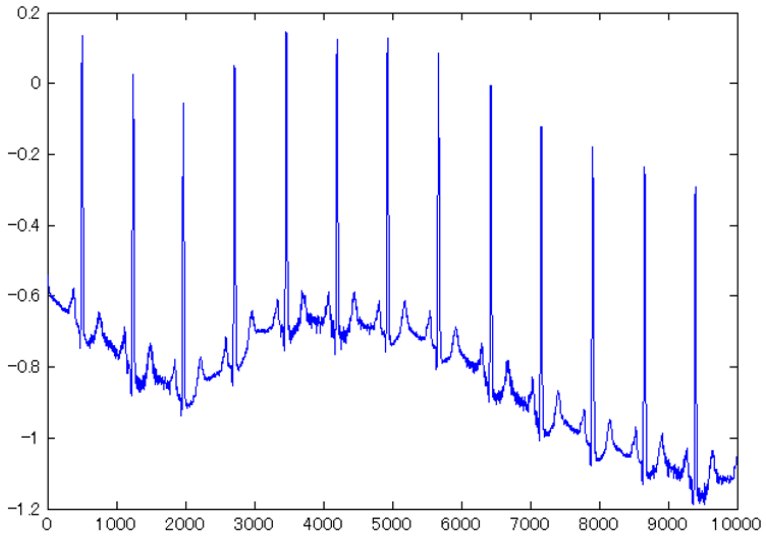


Figure. 3.27: Disease data (Sequence 12)

3.2 A New Discretization Method for ECG

We proposed a new discretization method for application to ECG data. In this step, only disease ECG data is used.

3.2.1 Normalization

Remove Trends

Normalization is executed on all sequential data in a preprocessing. After normalization, sequential data are converted scale from 0 to 1. In the new discretization, normalization method is same as proposed method. However, it is necessary to remove trends in ECG before normalization. This is because ECG indicates swinging heartbeats. Thus, it is difficult to extract specific pattern that have features from ECG data without remove trends. Because of above reason, removal trends is executed before the normalization. Figure.3.28 – 3.39 show normalized disease data after removal trends. In these figure, horizon tall axis is the length of data and vertical axis is the normalization scale.

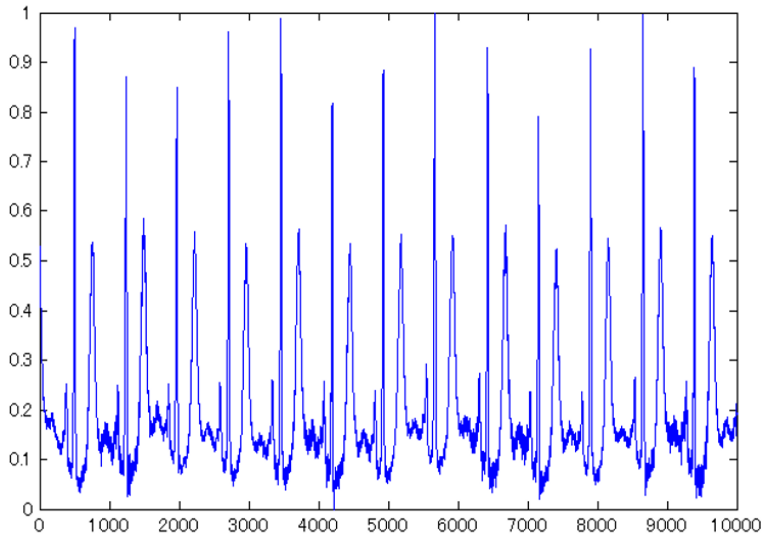


Figure. 3.28: Normalized disease data (Sequence 1)

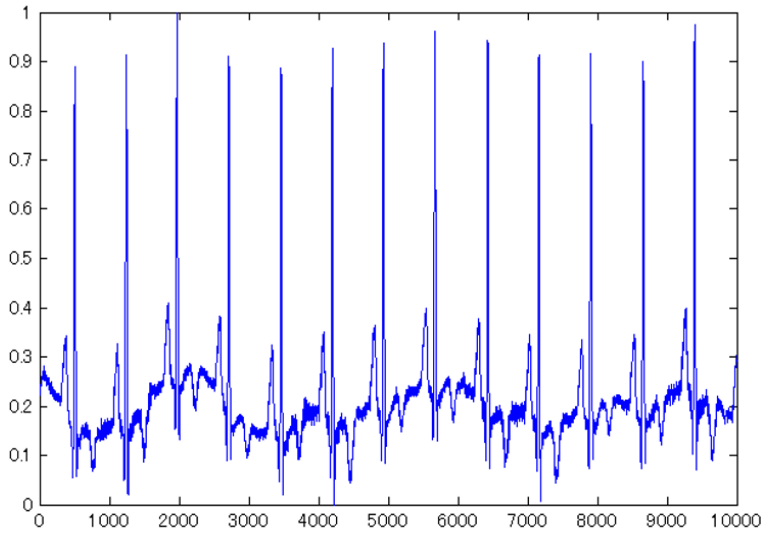


Figure. 3.29: Normalized disease data (Sequence 2)

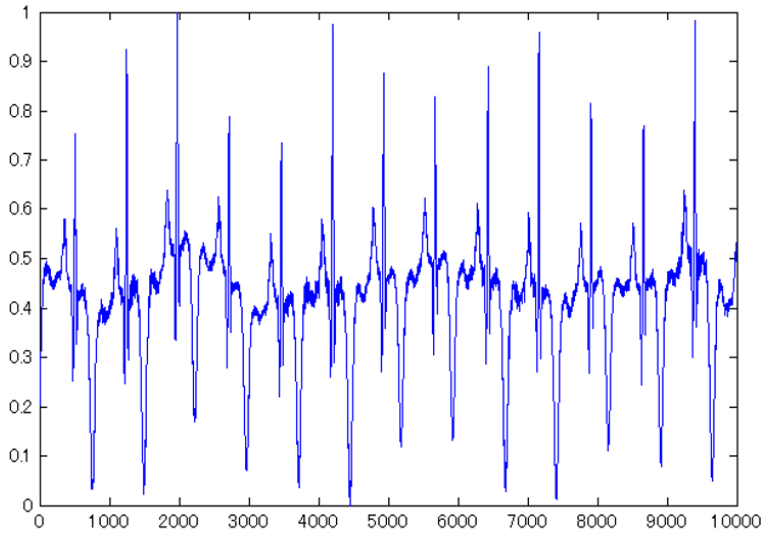


Figure. 3.30: Normalized disease data (Sequence 3)

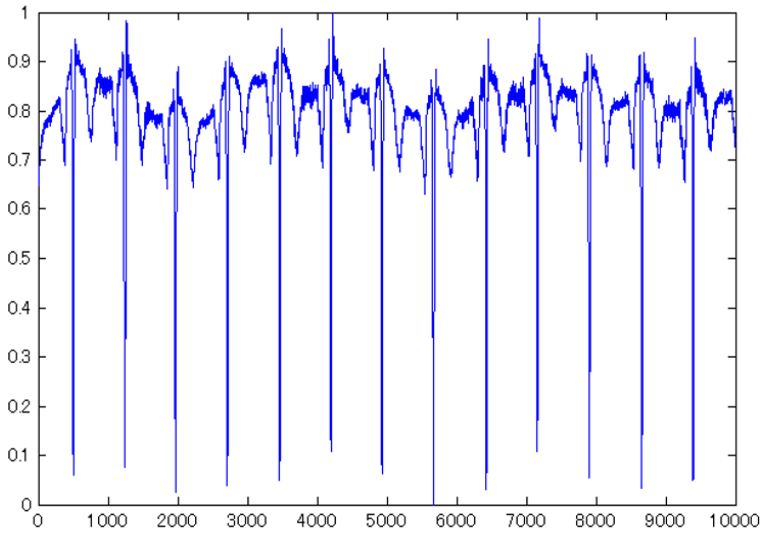


Figure. 3.31: Normalized disease data (Sequence 4)

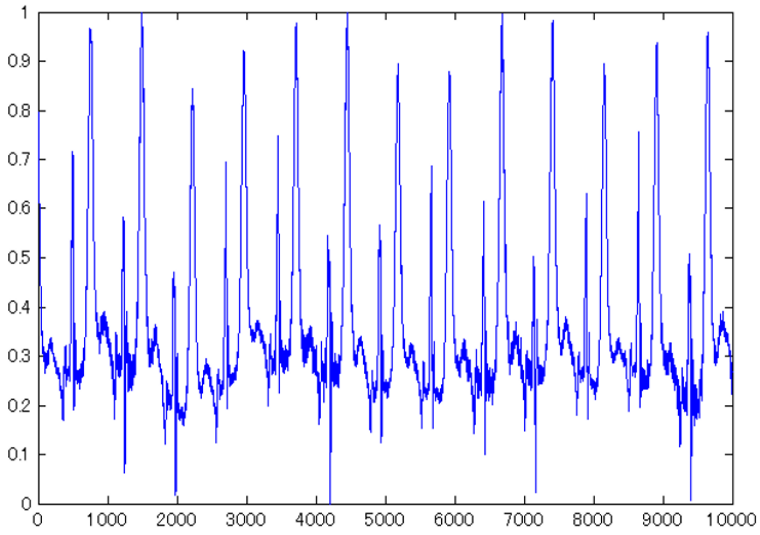


Figure. 3.32: Normalized disease data (Sequence 5)

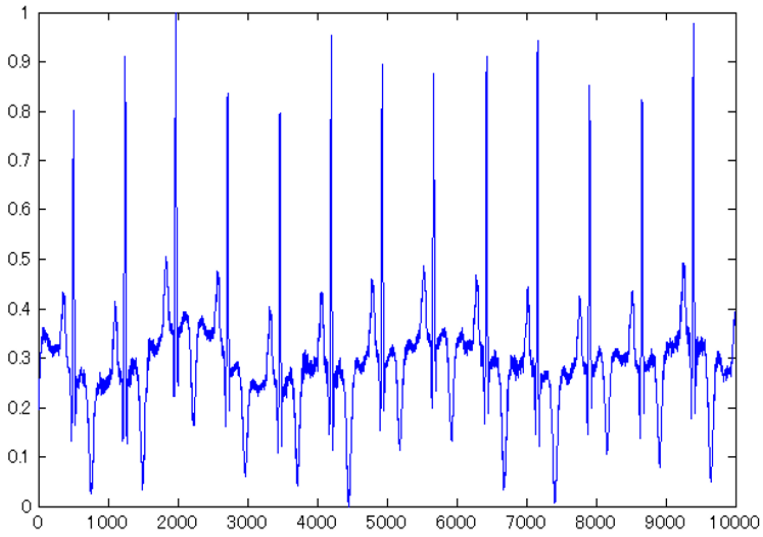


Figure. 3.33: Normalized disease data (Sequence 6)

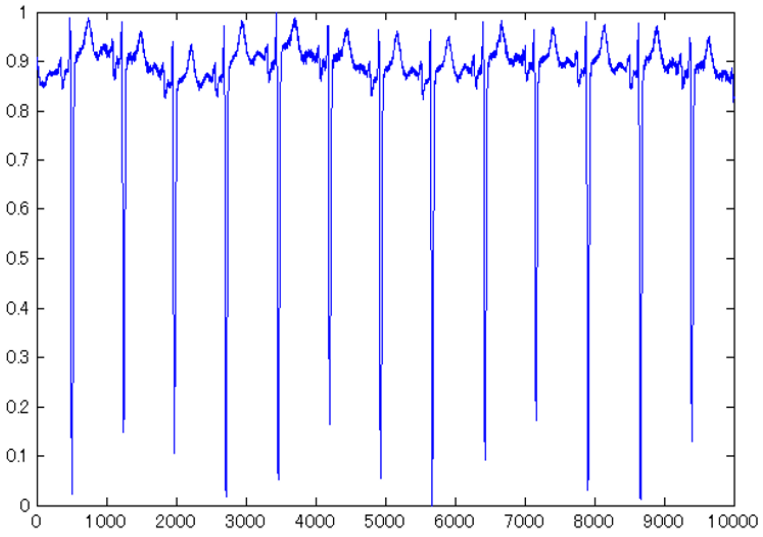


Figure. 3.34: Normalized disease data (Sequence 7)

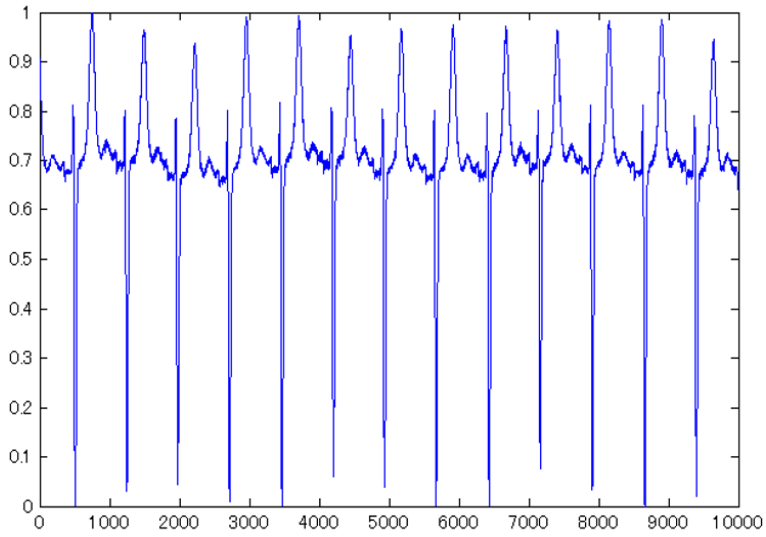


Figure. 3.35: Normalized disease data (Sequence 8)

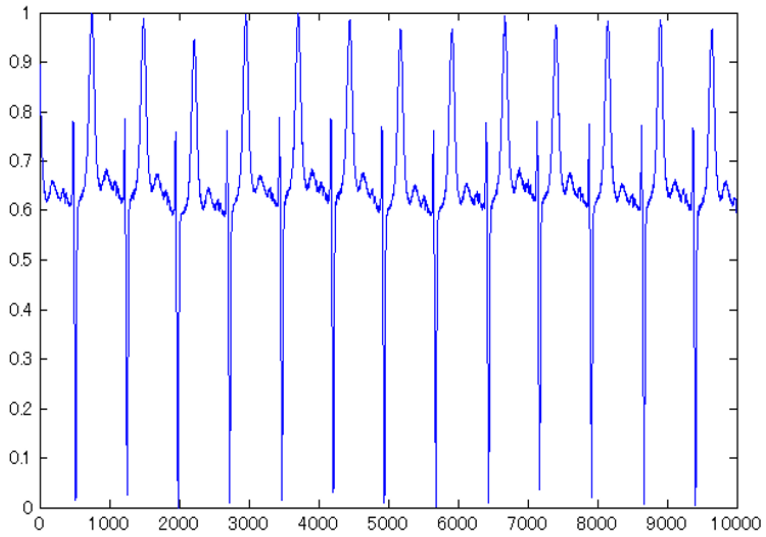


Figure. 3.36: Normalized disease data (Sequence 9)

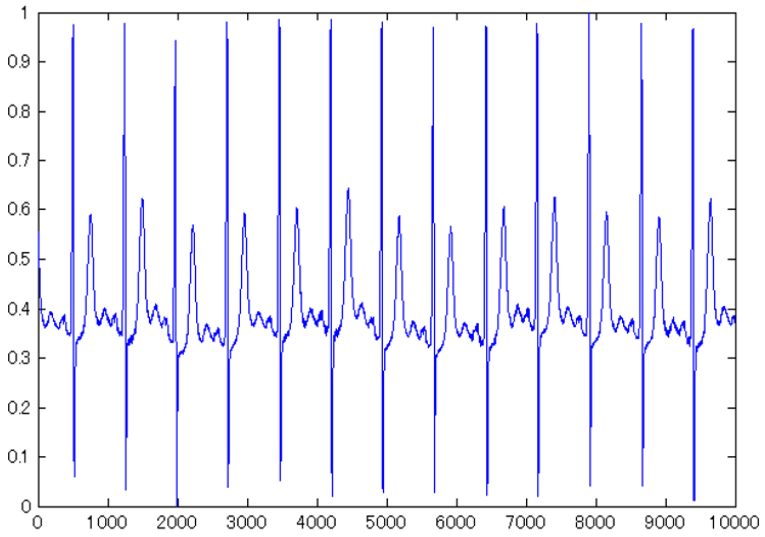


Figure. 3.37: Normalized disease data (Sequence 10)

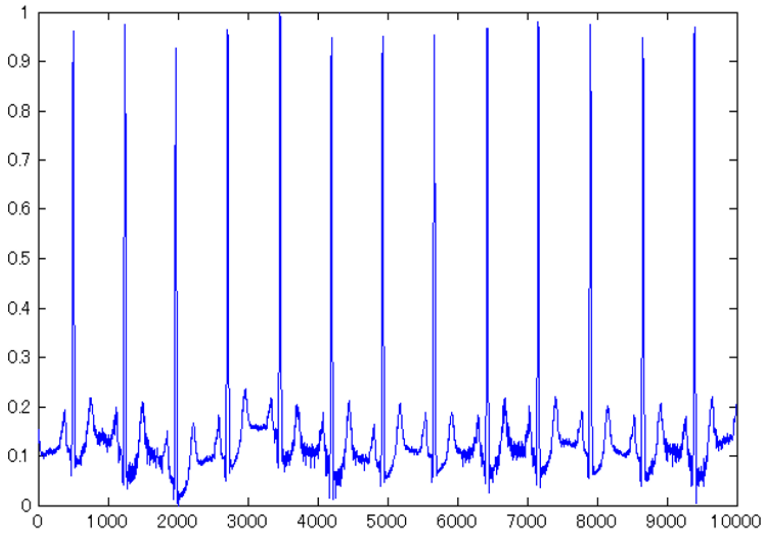


Figure. 3.38: Normalized disease data (Sequence 11)

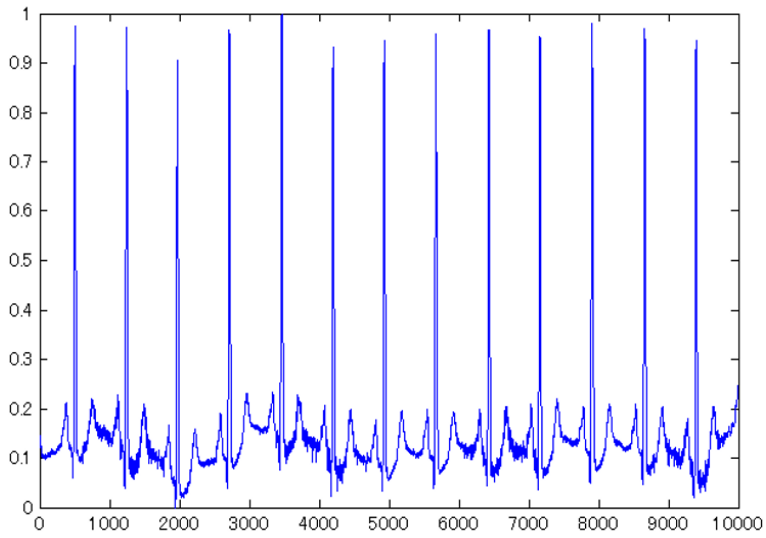


Figure. 3.39: Normalized disease data (Sequence 12)

3.2.2 Discretization

In the discretization, the range of normalized data (01) is divided at the D stages, and a discrete value from 0 to D-1 is allocated to each data. For application to real sequential data, discretization is specifically important step. This is due to the following reasons. First, ECG waveform changing drastically in section of QRS wave. Furthermore, it is possible to change input value of data by the setting number of D value in the step of discretization. D value is fixed by 50 in proposed method. This means that it is possible to lose important information from ECG because of fixed value. Therefore, in this study, we proposed a new discretization for ECG data.

Figure.3.40 shows the procedure of new discretization. In this new method, input is normalized ECG data. We select 250 points before R wave(peak of waveform) and 450 points after R wave from ECG data. Hence, the length of the one waveform is 701. We used square-root choice in order to decide bin number of histogram. Thus, bin number of histogram was set to 26. The histogram of each sequence of ECG is generated by 26 bin numbers(Figure.3.40(a)). Next, class numbers of each bin are calculated using square-root choice(Figure.3.40(b)). After that, ECG data is divided by the class number(Figure.3.40(c)). In this figure, ECG data is divided by 105 because all number of class is 105. Discretized ECG data is output of new discretization method.

As above steps, discretization are executed about twelve sequences of disease ECG data. These steps are explained in detail below.

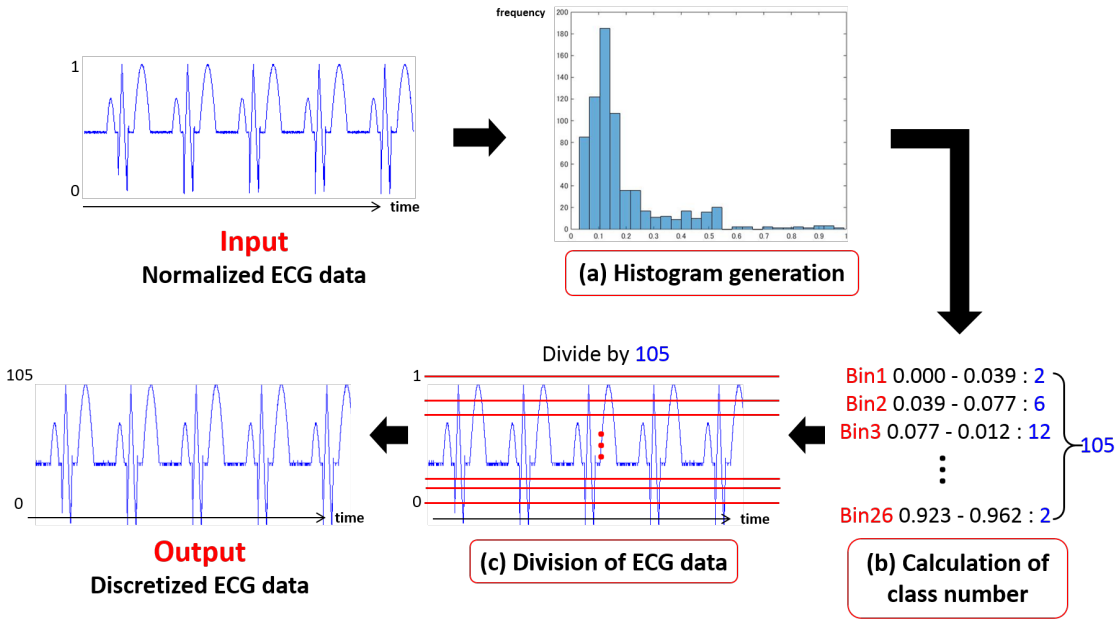


Figure. 3.40: Procedure of discretization

Histogram Generation

First, we generate histogram from normalized ECG data. We used square-root choice in order to decide bin number of histogram. The square-root choice were calculated as follows equation.

$$k = \sqrt{n} \tag{3.1}$$

Here, k means bin number, and n means the number of data point. We select 250 points before R wave(peak of waveform) and 450 points after R wave from ECG data. Hence, the length of the one waveform is 701. From this equation, bin number of histogram was set to 26. Consequently, the histogram of each sequence of ECG is generated by 26 bin numbers(Figure.3.40(a)).

Figure.3.41 – Figure.3.52 show each histogram of twelve sequences of disease data. In these figure, horizon tall axis is the normalization scale and vertical axis is data points of each bin.

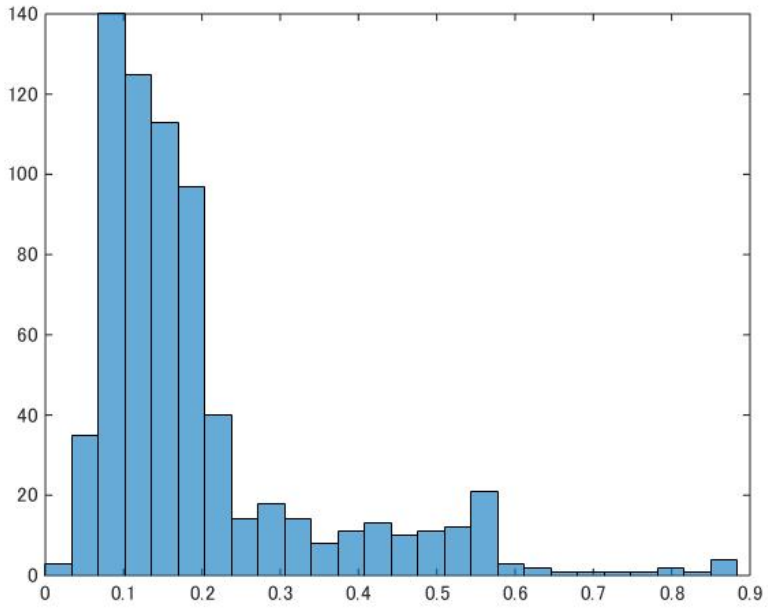


Figure. 3.41: Histogram (Sequence 1)

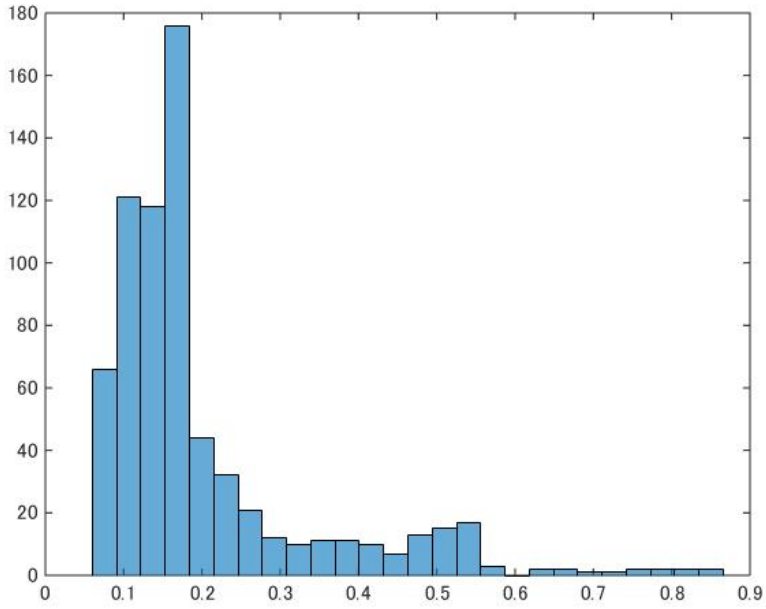


Figure. 3.42: Histogram (Sequence 2)

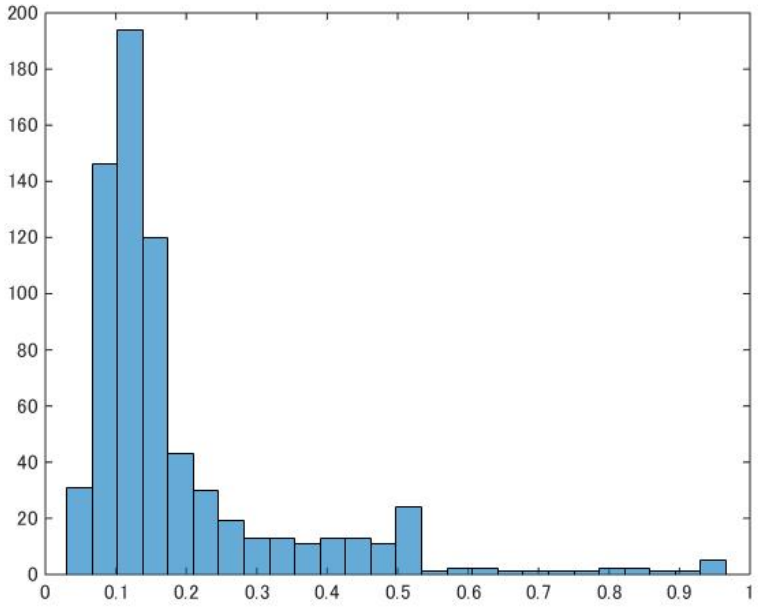


Figure. 3.43: Histogram (Sequence 3)

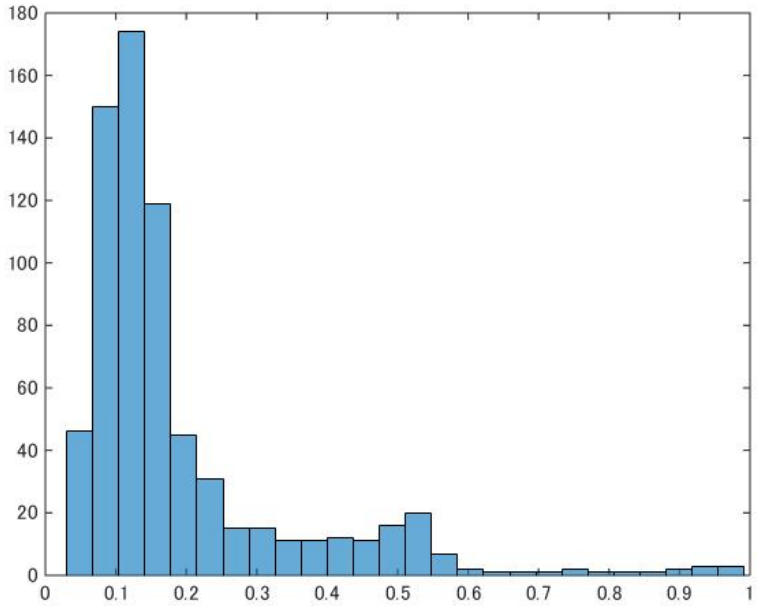


Figure. 3.44: Histogram (Sequence 4)

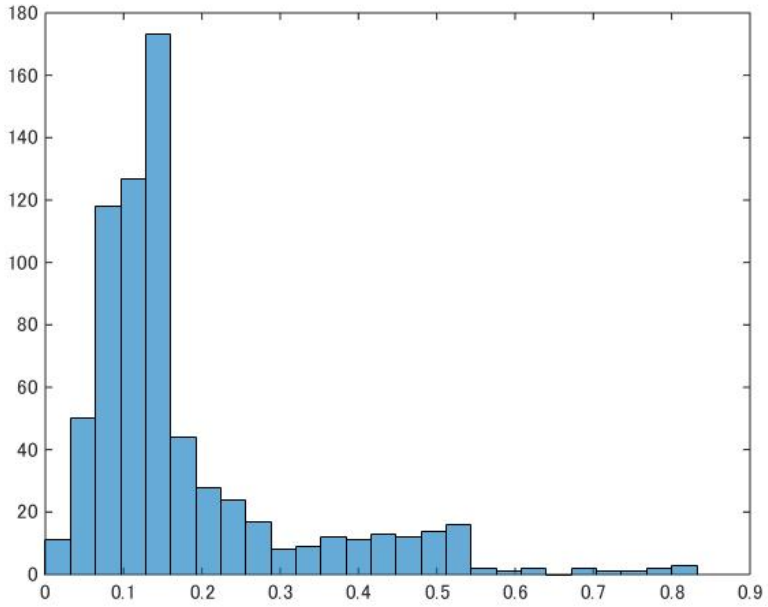


Figure. 3.45: Histogram (Sequence 5)

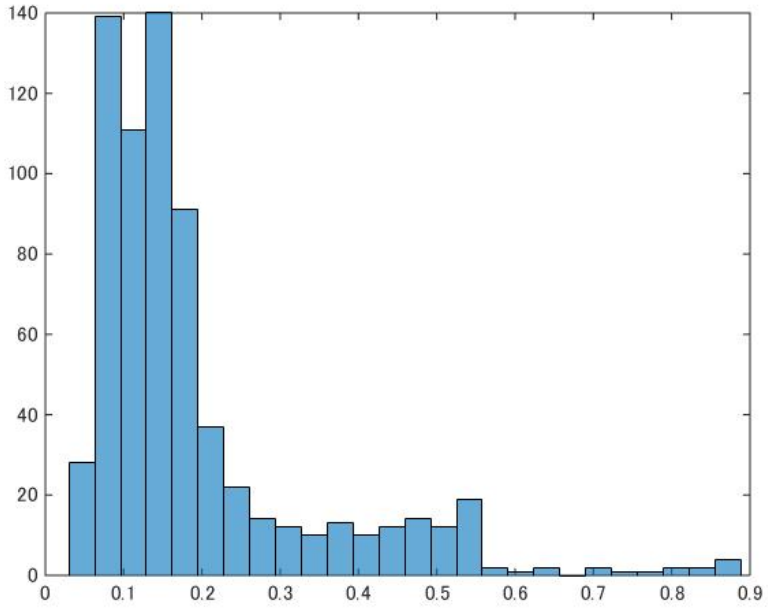


Figure. 3.46: Histogram (Sequence 6)

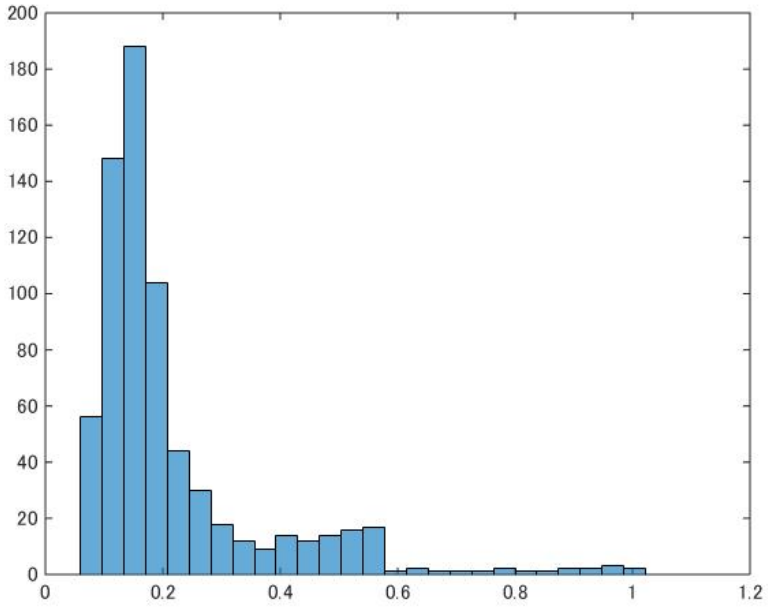


Figure. 3.47: Histogram (Sequence 7)

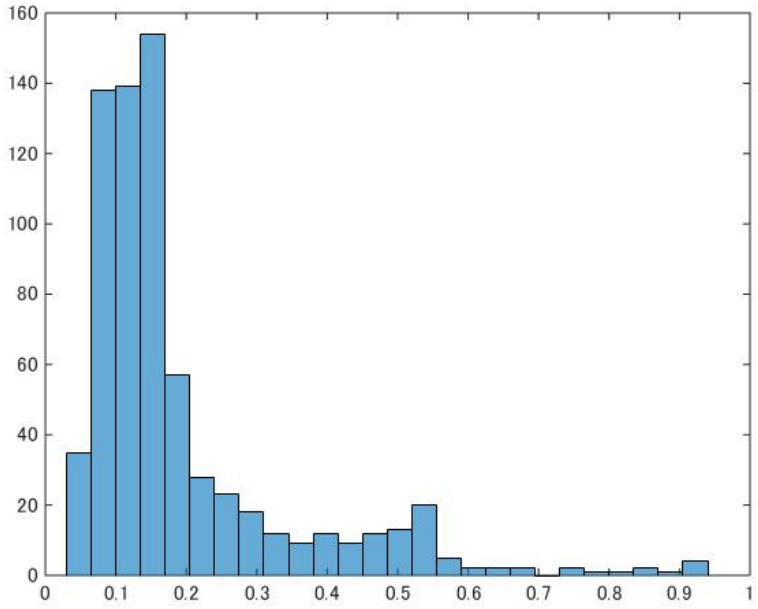


Figure. 3.48: Histogram (Sequence 8)

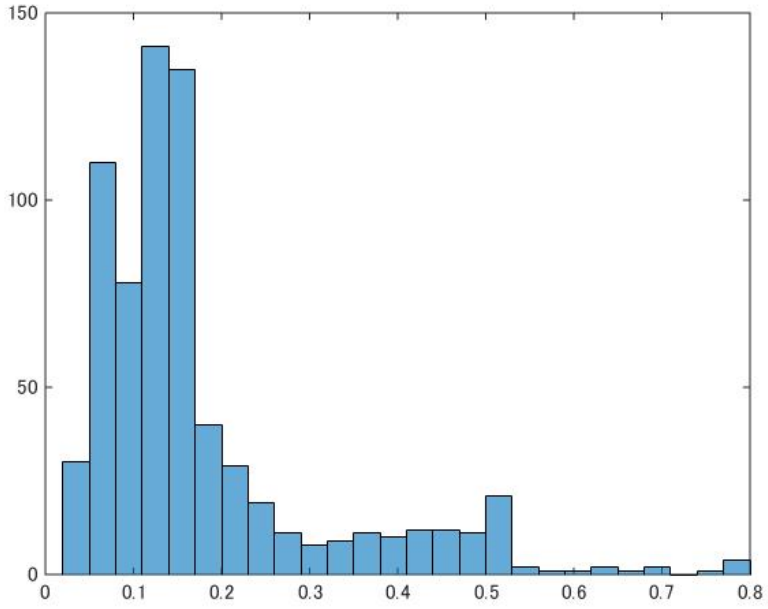


Figure. 3.49: Histogram (Sequence 9)

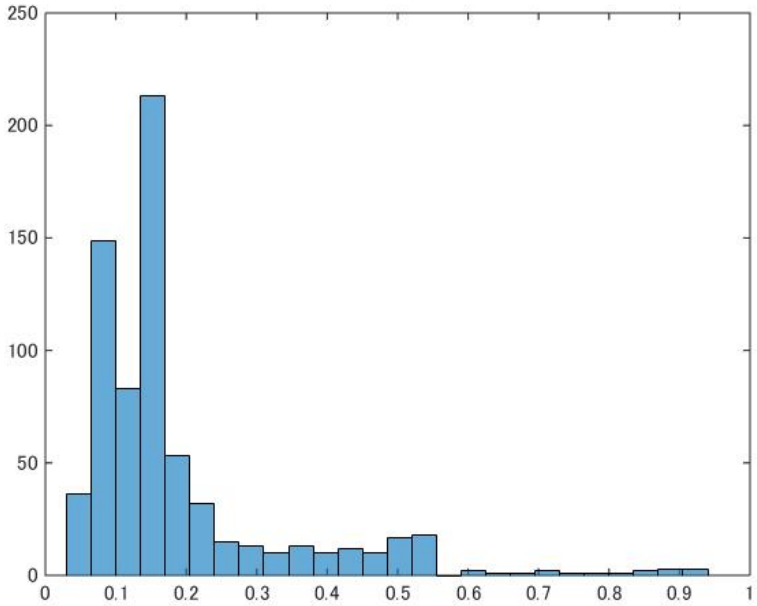


Figure. 3.50: Histogram (Sequence 10)

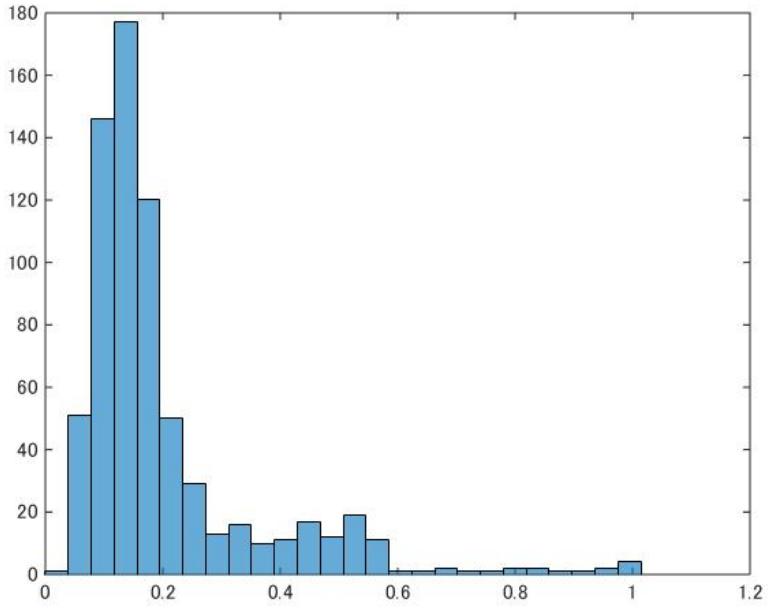


Figure. 3.51: Histogram (Sequence 11)

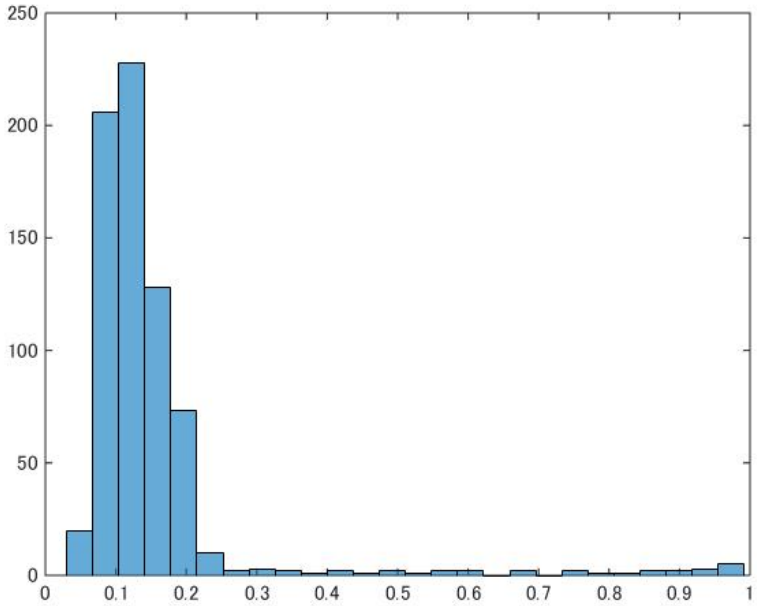


Figure. 3.52: Histogram (Sequence 12)

Calculation of Class Number

Next, class numbers of each bin are calculated using square-root choice again(Figure.3.40(b)). The equation of square-root choice is explained above step histogram generation. Table.3.1 shows the class number of 26 bins on each sequence of discretized disease data. The range of bins that have large data points was divided by large class number. In contrast, the range of bins that have small data points was divided by small class number.

Table. 3.1: Class number of distribution on discretized disease data

	Class number of distribution	Total number
Sequence 1	2 6 12 11 11 10 6 4 4 4 3 3 4 3 3 3 5 2 1 1 1 1 1 1 1 2	105
Sequence 2	3 3 6 9 16 10 12 4 5 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2	89
Sequence 3	2 5 3 4 2 3 3 4 6 5 6 13 15 7 5 3 1 2 2 2 1 1 1 1 1 2	100
Sequence 4	2 2 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 4 8 10 13 9 14 6 2 3	86
Sequence 5	2 2 2 3 5 12 14 10 6 4 3 4 5 4 2 2 2 2 3 2 3 3 2 3 4 1	105
Sequence 6	5 4 5 6 4 6 12 13 11 5 5 3 2 1 1 1 0 1 1 1 1 1 1 1 1 2	94
Sequence 7	1 2 2 1 1 2 2 1 1 1 2 1 1 1 2 1 1 1 1 2 2 2 15 18 10 2	76
Sequence 8	2 2 2 2 1 2 2 1 1 2 1 2 1 1 1 2 2 3 19 12 6 6 4 4 4 6	91
Sequence 9	1 2 2 2 2 1 1 2 1 2 2 1 2 2 2 3 15 15 6 5 5 4 4 4 4 5	95
Sequence 10	2 1 2 1 2 1 2 2 14 17 8 5 4 4 5 6 2 1 1 2 1 0 2 1 2 2	90
Sequence 11	9 14 15 10 8 1 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 2 2	76
Sequence 12	4 14 15 11 9 3 1 2 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 2 2	77

Division of ECG Data

ECG data was divided by the class number(Figure.3.40(c)) as above table. For example, sequence 12 was divided by 26 bins and each bin was divided by 4, 14, 15, 11, 9, 3, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 2, and 2 in normalized range of 0-1. Discretized sequences of disease ECG data are output of new discretization method. Figure.3.53 – Figure.3.64 show the discretized disease data of each sequence. In these figure, horizon tall axis is the length of disease data(data points) and vertical axis is the discrete value.

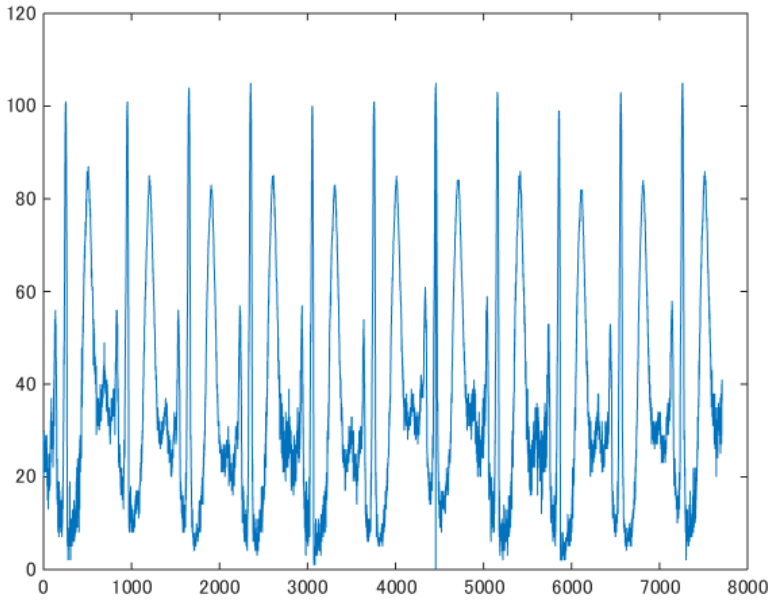


Figure. 3.53: Discretized disease data (Sequence 1)

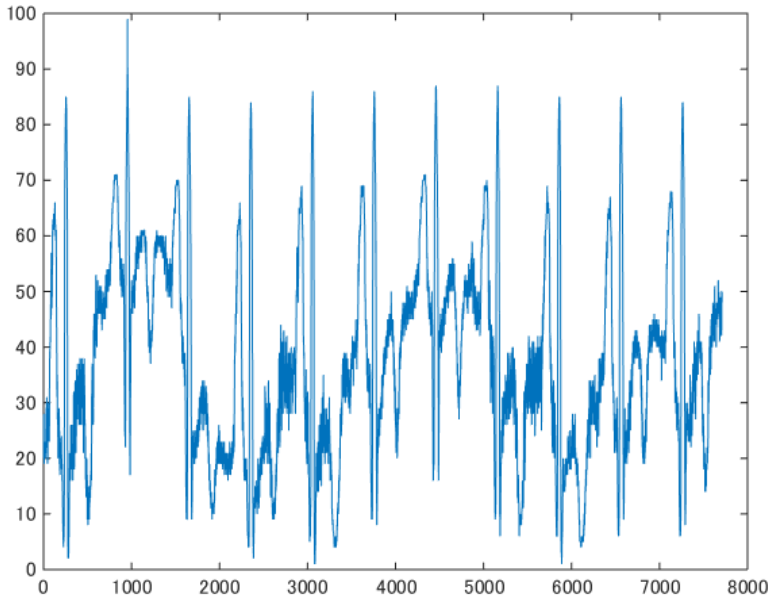


Figure. 3.54: Discretized disease data (Sequence 2)

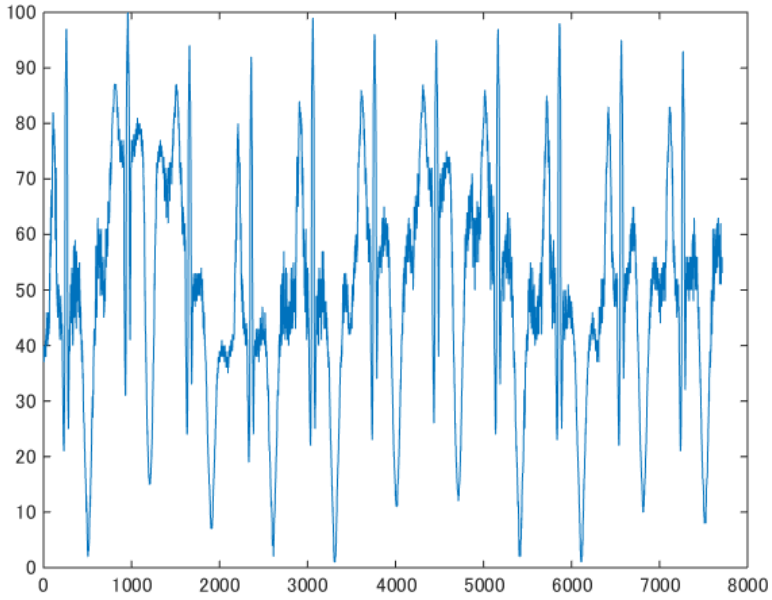


Figure. 3.55: Discretized disease data (Sequence 3)

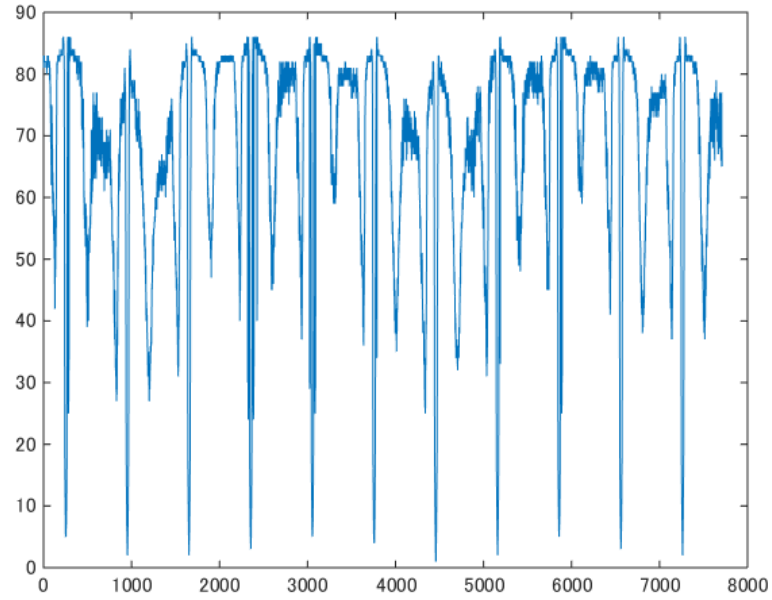


Figure. 3.56: Discretized disease data (Sequence 4)

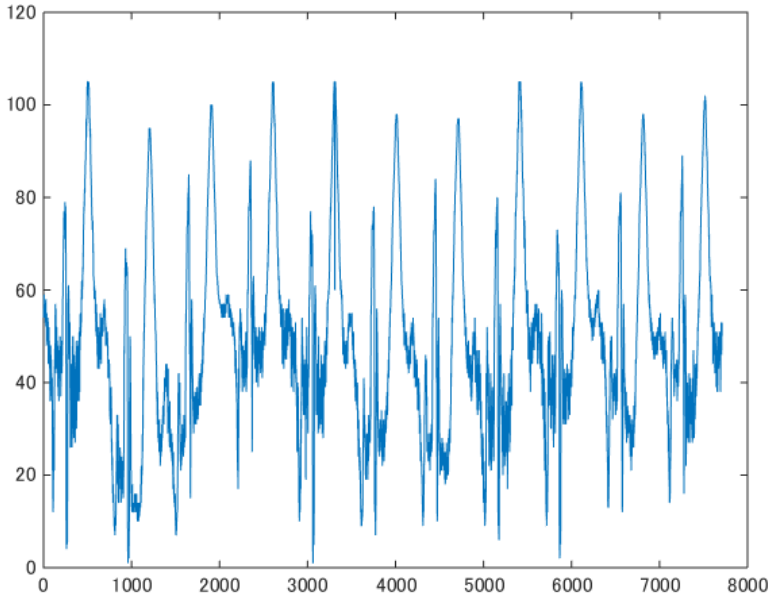


Figure. 3.57: Discretized disease data (Sequence 5)

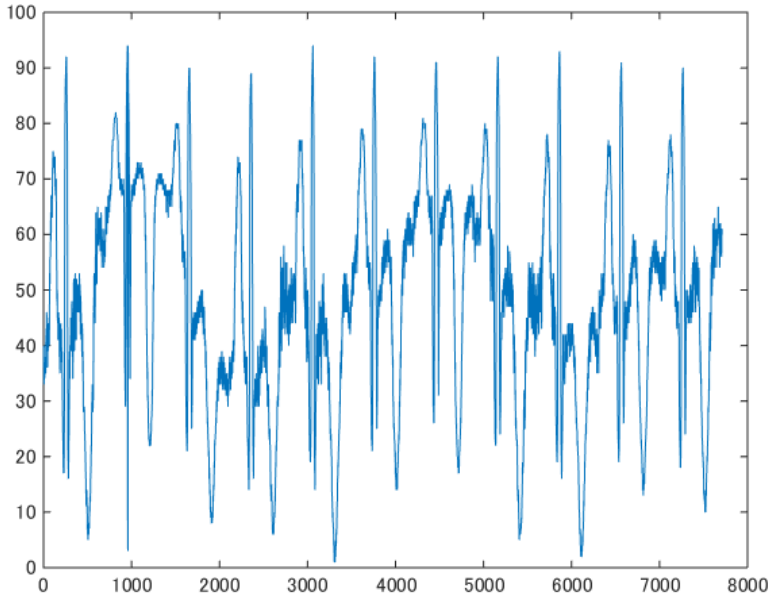


Figure. 3.58: Discretized disease data (Sequence 6)

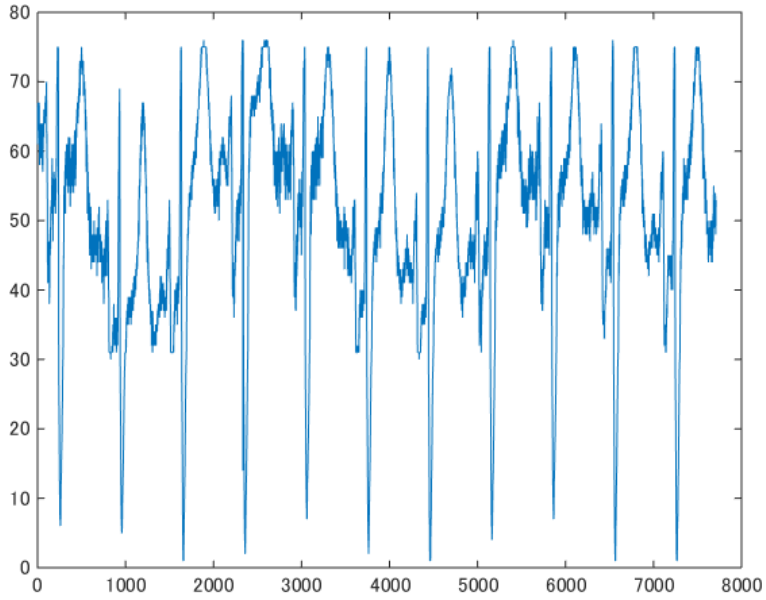


Figure. 3.59: Discretized disease data (Sequence 7)

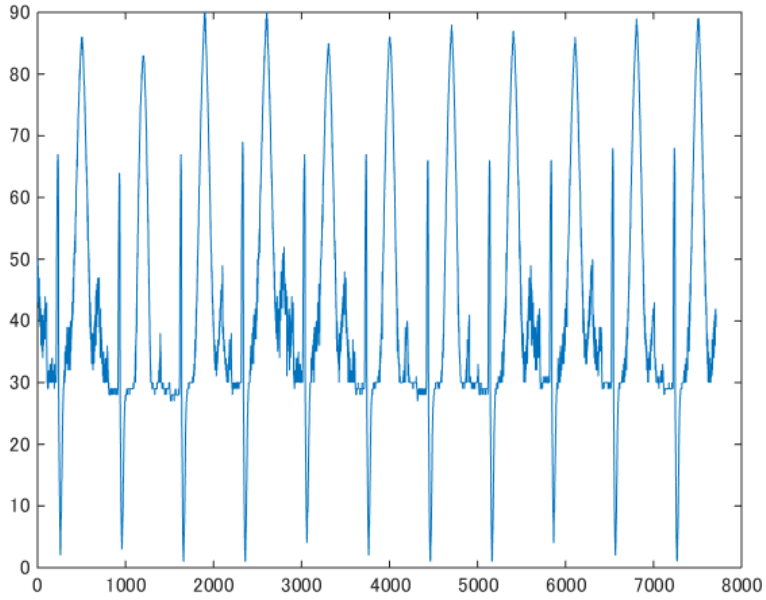


Figure. 3.60: Discretized disease data (Sequence 8)

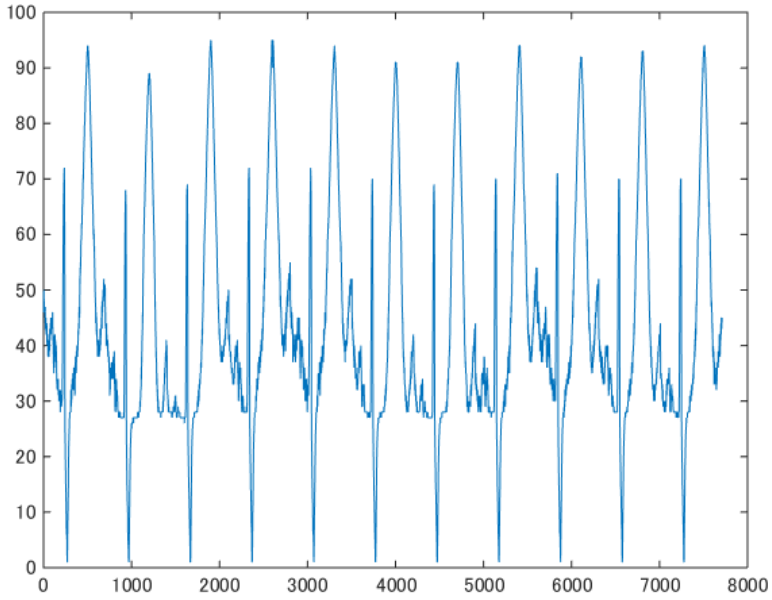


Figure. 3.61: Discretized disease data (Sequence 9)

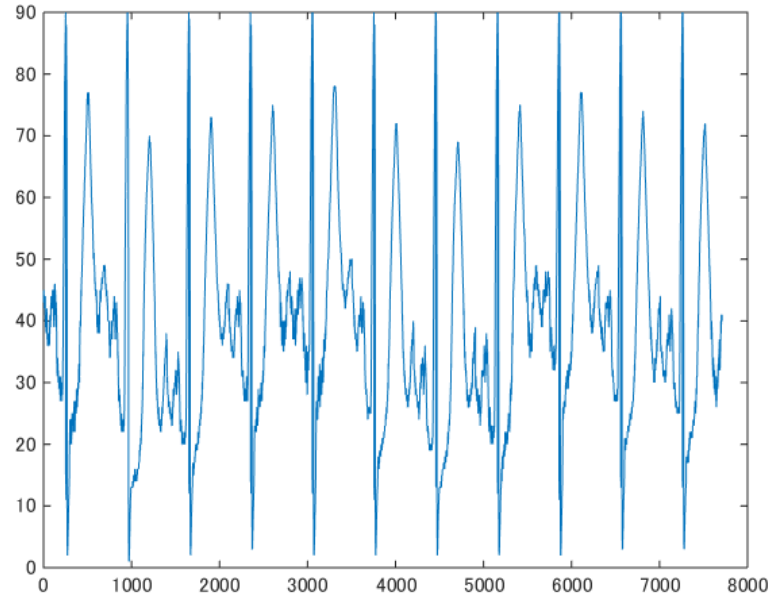


Figure. 3.62: Discretized disease data (Sequence 10)

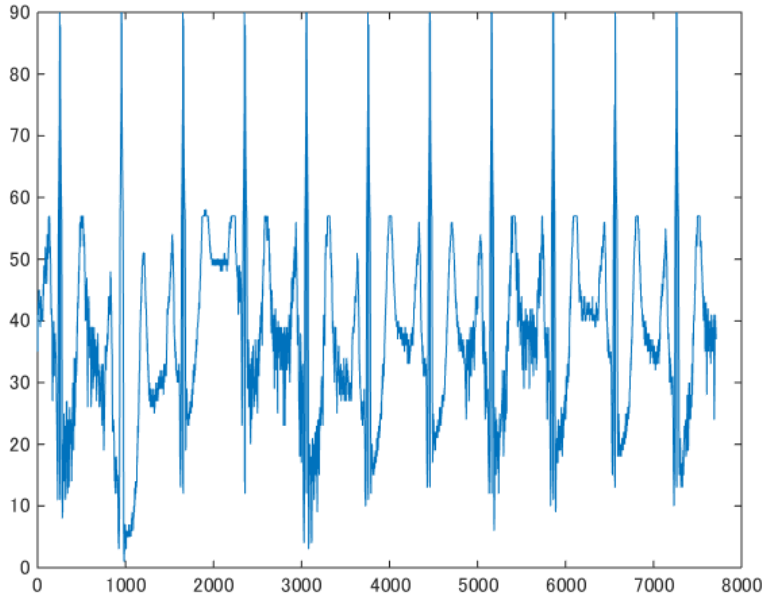


Figure. 3.63: Discretized disease data (Sequence 11)

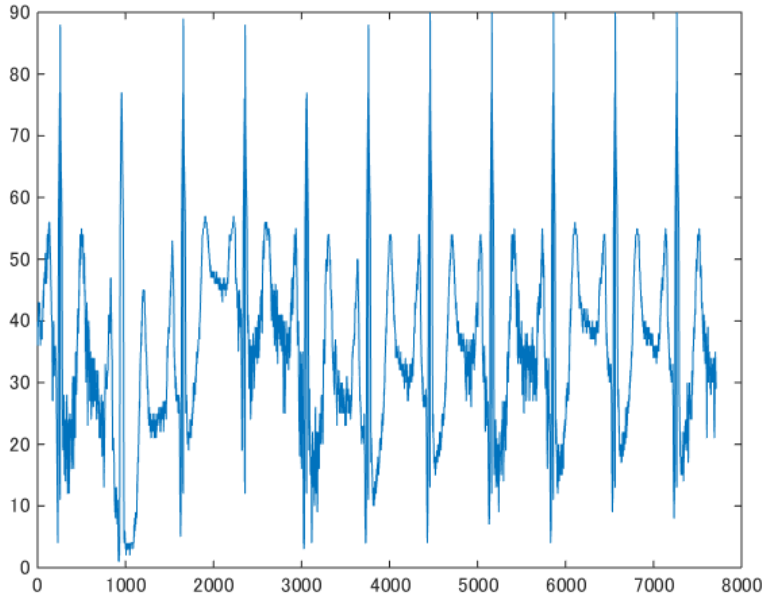


Figure. 3.64: Discretized disease data (Sequence 12)

3.3 Experiment

The proposed method based on new discretization was evaluated for extraction accuracy using disease ECG data. The goal of this experiment is to compare the performance of the previous and proposed methods using disease ECG data for extracting abnormal waveform.

3.3.1 Type of Disease ECG Data

In this experiment, disease ECG data was used for detection abnormal waveform. As described in section 3.1.2 Disease Data, this disease data is myocardial infarction(MI). Table.3.2 shows the types of myocardial infarction and the sequences indicating abnormal waveform by MI types. The type of myocardial infarction of disease data is an inferior. Therefore, there exist abnormal Q and ST wave in the area of sequences, II, III, ${}_aV_F$, V_5, and V_6. We can see that abnormal Q and ST wave in the sequences, II, III, ${}_aV_F$, V_5, and V_6 compare with healthy ECG data visually.

Table. 3.2: Types of myocardial infarction and the sequences indicating abnormal Q and ST wave

Types of myocardial infarction	Sequences indicating abnormal Q and ST wave
Septal	V_1, V_2
Anterior	V_3, V_4
Anteroseptal	V_1, V_2, V_3, V_4
Anterolateral	I, V_3, V_4, V_5, V_6, ${}_aV_L$
Extensive anterior	I, V_1, V_2, V_3, V_4, V_5, V_6, ${}_aV_L$
Inferior	II, III, ${}_aV_F$, V_5, V_6
Lateral	I, ${}_aV_L$, V_5, V_6
Posterior	V_7, V_8, V_9
RV	II, III, ${}_aV_F$, $V1$, V_4

3.3.2 Parameter Setting

For frequent pattern extraction, the window width w values was fixed at 3 and the minimum number of occurrences θ were set to natural numbers ≥ 5 in increments of 5.

For closed itemset mining, *minsup* values were set to 3. For discretization, in the new method, the discrete values of each sequence was set to 105, 89, 100, 86, 105, 94, 76, 91, 95, 90, 76, and 77. In the previous discretization method, discrete values were set to fixed value same as new discretization. In the new discretization method, the range of bins that have large data points was divided by large class number. In contrast, the range of bins that have small data points was divided by small class number. However, In the previous discretization method, the all range of bins was divided by equally even same discrete value with proposed method.

3.3.3 Extraction Accuracy of Linkage Pattern

The extraction accuracies of the embedded linkage patterns for the previous and proposed discretization methods were compared using the above disease ECG data. In this experiment, the extraction accuracy means that how many abnormal waveform was detected from disease ECG data. Therefore, only sequences, II, III, ${}_aV_F$, V_5, and V_6 indicating abnormal Q and ST wave in inferior myocardial infarction was used for extraction accuracy. Precision, recall, and F-measure were used as evaluation indexes. These indexes were calculated as follows.

$$\begin{aligned} Precision &= CDP/DDP \\ Recall &= CDP/EDP \\ F\text{-}measure &= 2 * Precision * Recall/Precision + Recall \end{aligned} \tag{3.2}$$

Here, CDP is the number of data points in the correctly detected areas of the abnormal waveform, DDP is the number of data points in the areas of the abnormal waveform detected by the method, and EDP is the number of data points in the abnormal waveform.

3.4 Results

3.4.1 Extraction Accuracy

Visualization of Extraction Accuracy

Figure.3.68 – Figure.3.72 are graphs of extracted linkage patterns in different θ when the previous discretization method was applied to the disease data. Figure.3.73 is graph of extracted linkage patterns when the proposed discretization method was applied to the healthy data. Figure.3.74 – Figure.3.77 are graphs of extracted linkage patterns in different θ when the proposed discretization method was applied to the disease data. In this figures, only these sequences II, III, ${}_aV_F$, V_5, and V_6 are indicated because the type of myocardial infarction of disease data is an inferior. Blue part shows the extracted abnormal waveform and yellow part shows the linkage pattern was extracted. As we can see, the abnormal waveform is detected by both of previous and proposed discretization method. In contrast, the proposed method shows that the abnormal waveform is more detected than previous method for all θ. This is because the previous method discretizes the disease ECG data using equal discrete value. However, the proposed method discretizes the disease ECG data using different discrete value by the number of data point in each class. This means that the disease ECG data was discretized more correctly by the proposed method than the previous method.

Extraction accuracy using indexed are explained in below.

Extraction Accuracy based on Indexes

Figure.3.65 – Figure.3.67 are graphs of precision, recall, and F-measure in different θ when the previous and proposed discretization methods were applied to the disease data. In these graph, horizon tall axis is the minimum number of occurrences θ and vertical axis is the score. From these graphs, the proposed method demonstrates high extraction accuracy for all θ compared previous method. This means that the abnormal waveform was extracted suitably by new discretization method. In addition, precision of both of previous and proposed discretization method tends to increase with increasing θ Precision shows the correctness of extraction accuracy and the minimum number of occurrences θ is the minimum number of frequent patterns to be extracted. Therefore, large θ value means strict condition to extract linkage pattern. From the above reasons, precision indicates high score when θ is large value. As a result, abnormal waveform of disease data was detected appropriately using the linkage pattern mining method with proposed discretization.

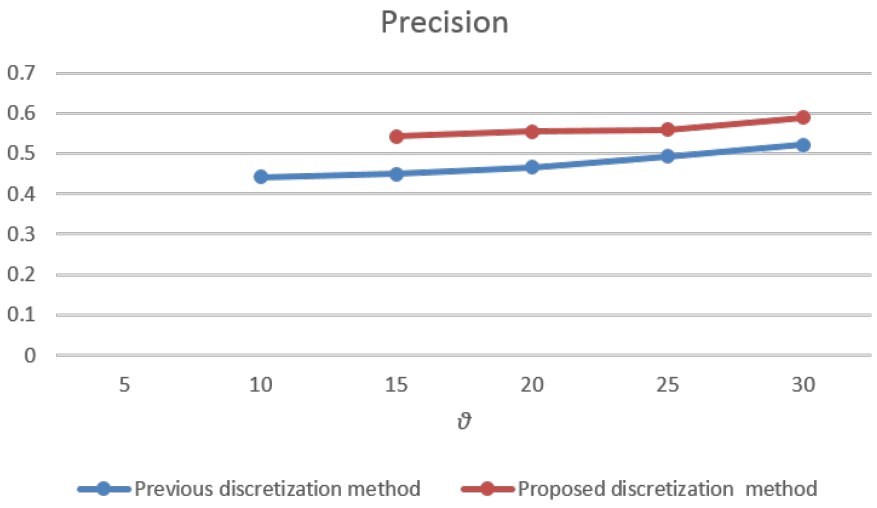


Figure. 3.65: Precision of previous and proposed discretization method

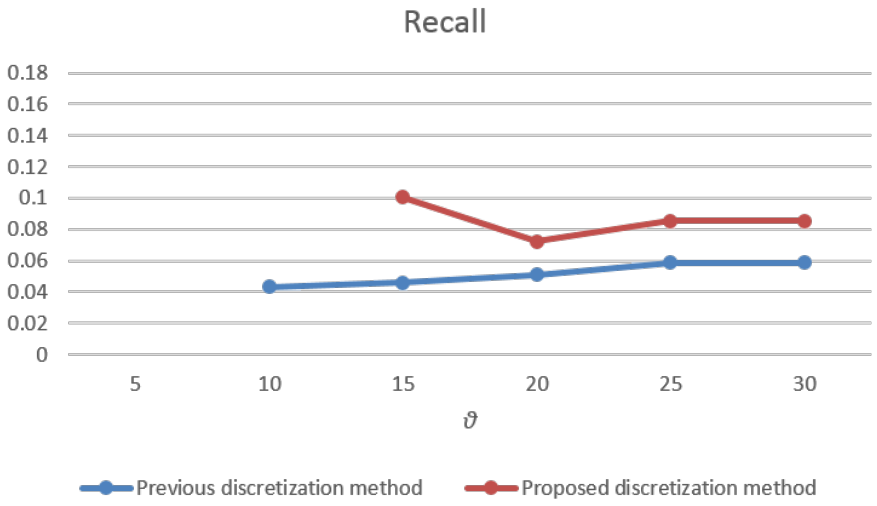


Figure. 3.66: Recall of previous and proposed discretization method

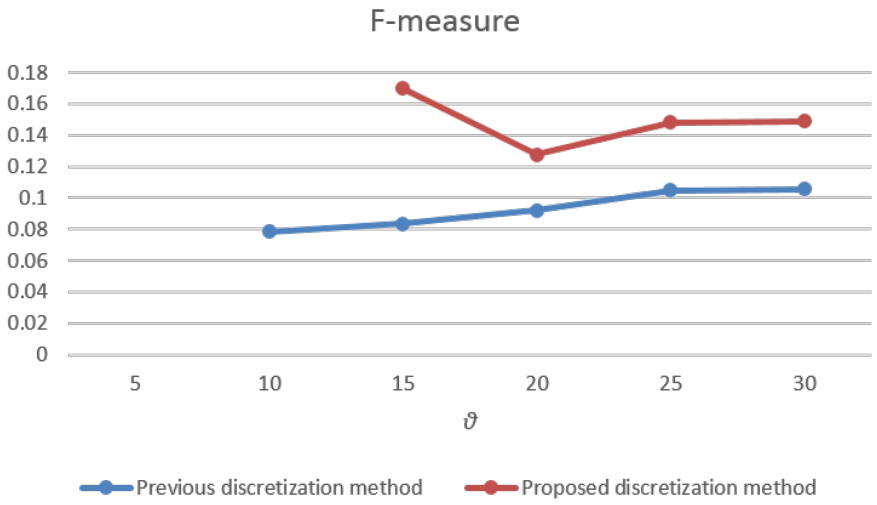


Figure. 3.67: F-measure of previous and proposed discretization method

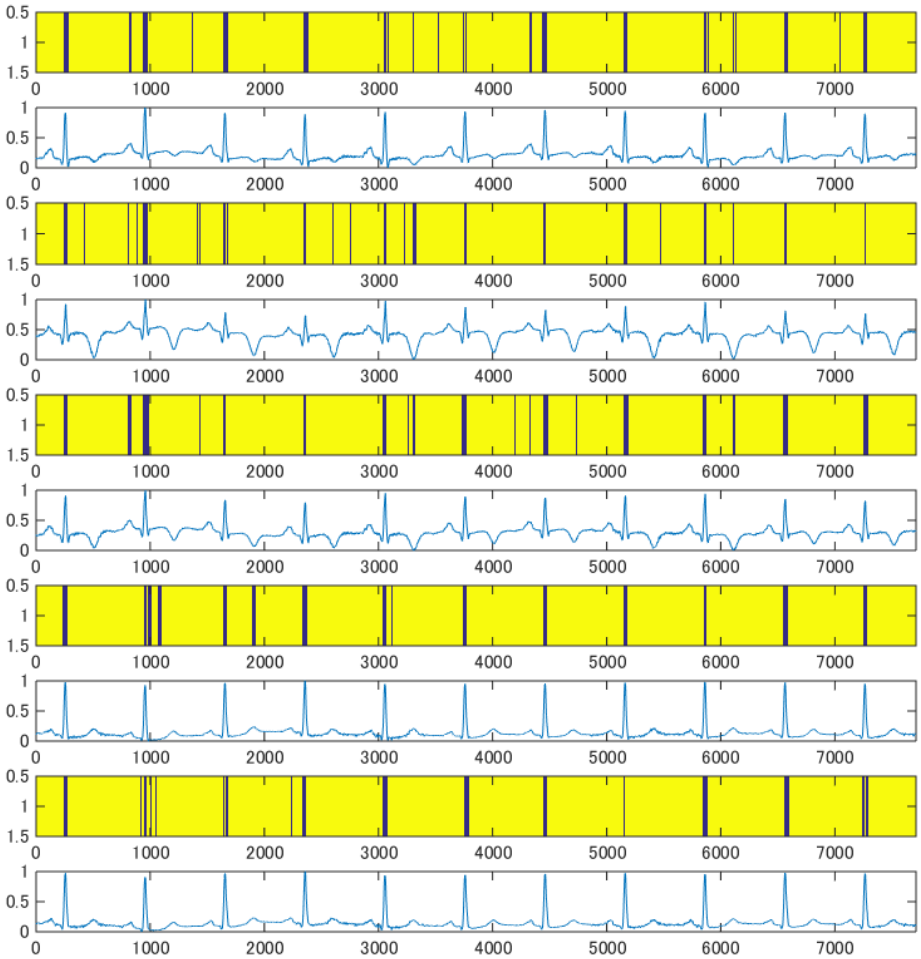


Figure. 3.68: Extraction accuracy of previous discretization method (θ =10)

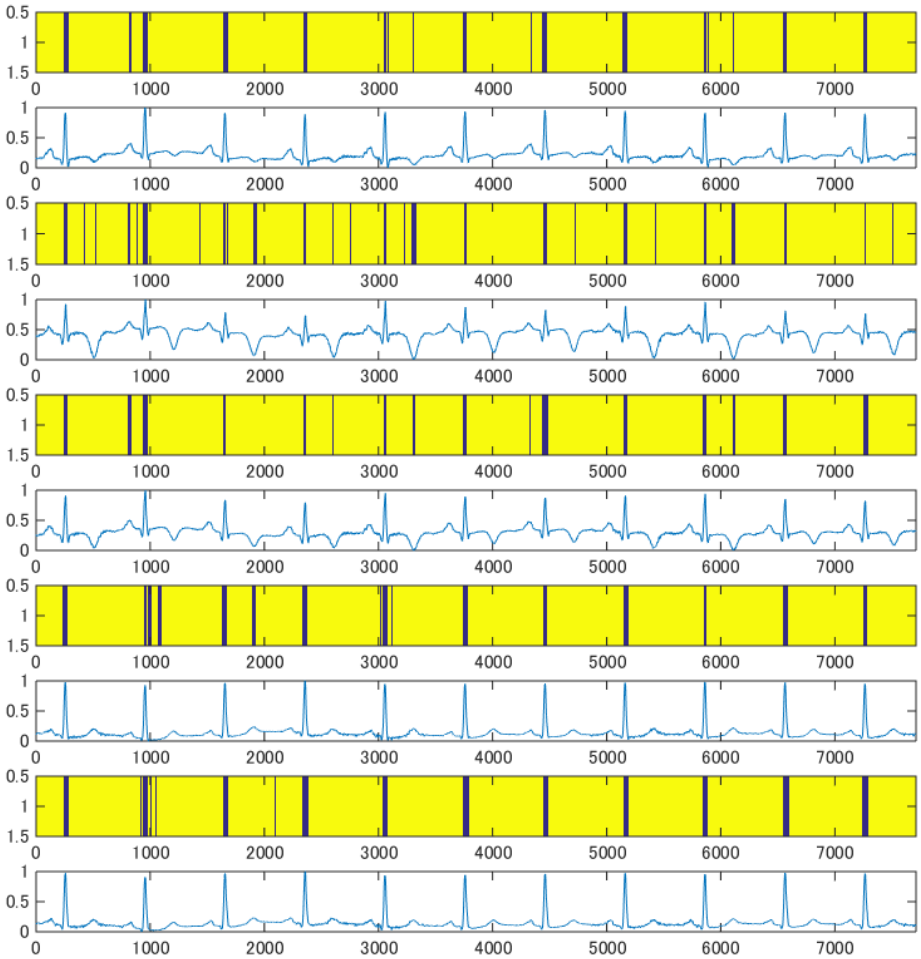


Figure. 3.69: Extraction accuracy of previous discretization method (θ =15)

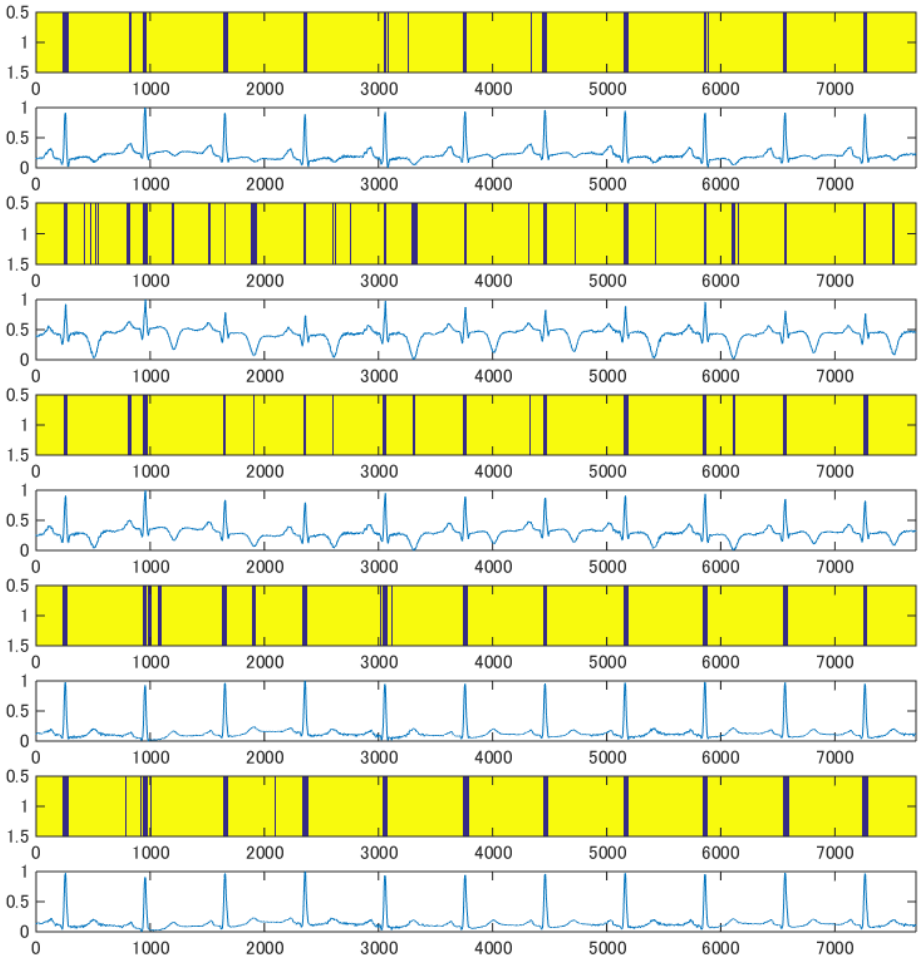


Figure. 3.70: Extraction accuracy of previous discretization method (θ =20)

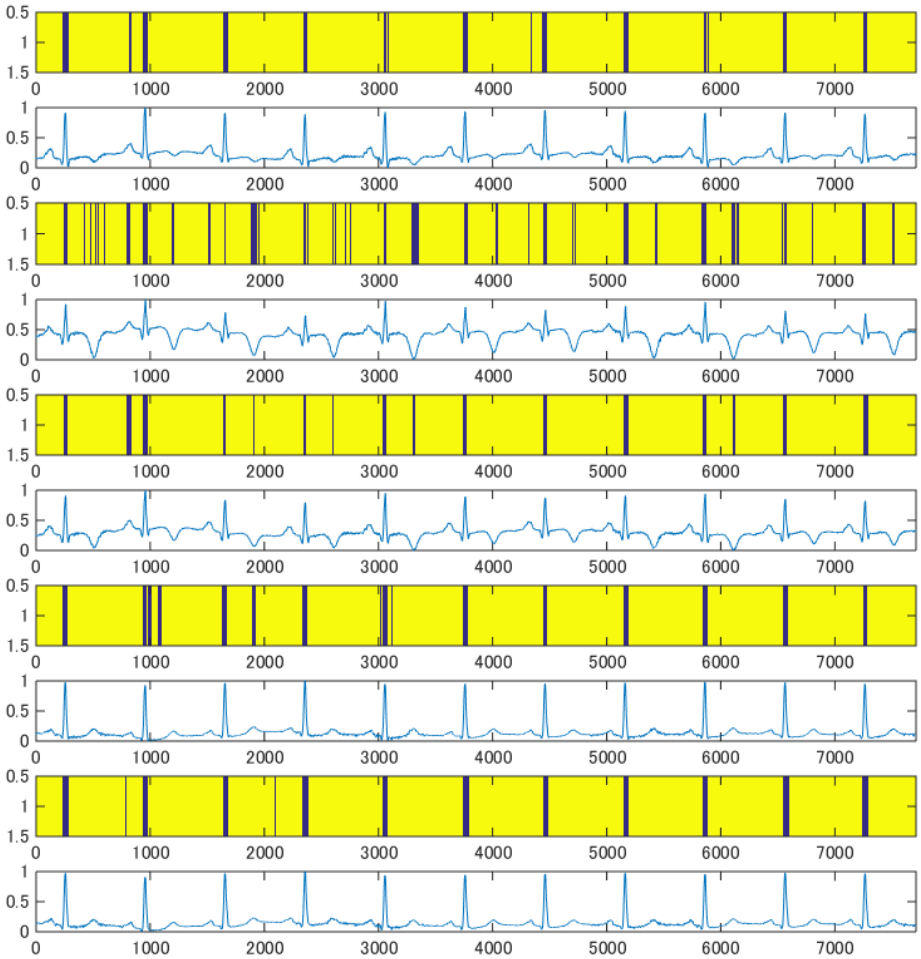


Figure. 3.71: Extraction accuracy of previous discretization method (θ =25)

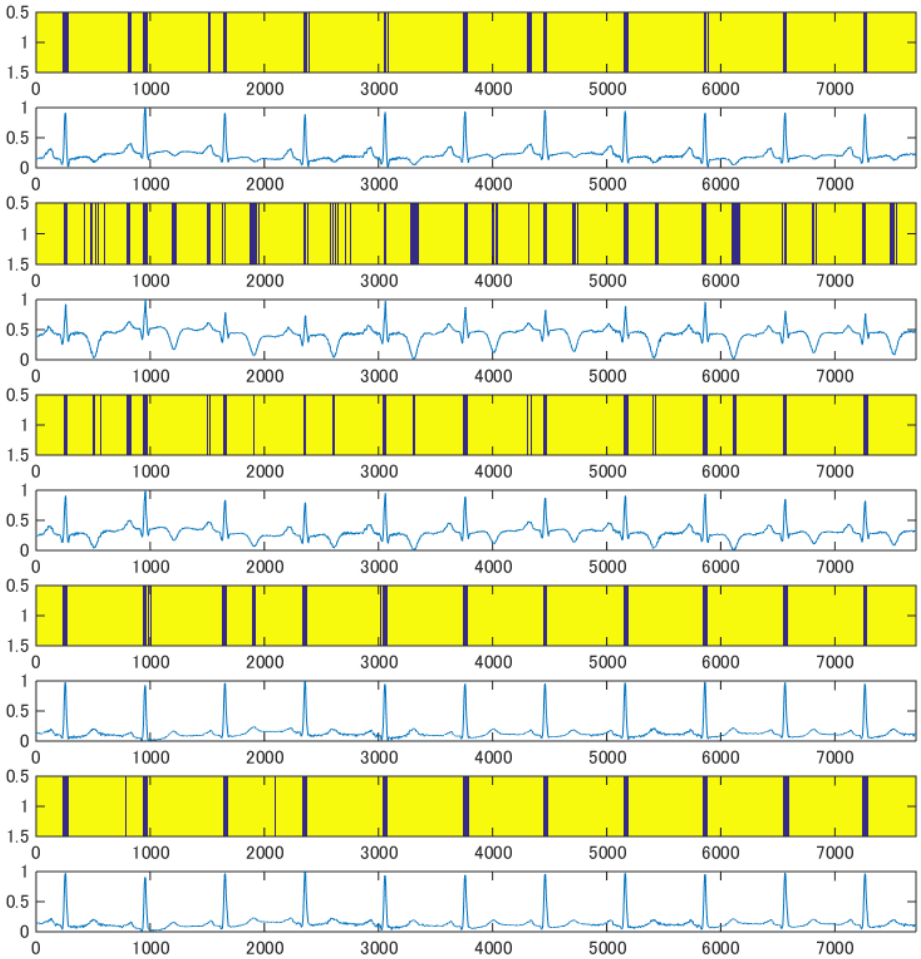


Figure. 3.72: Extraction accuracy of previous discretization method (θ =30)

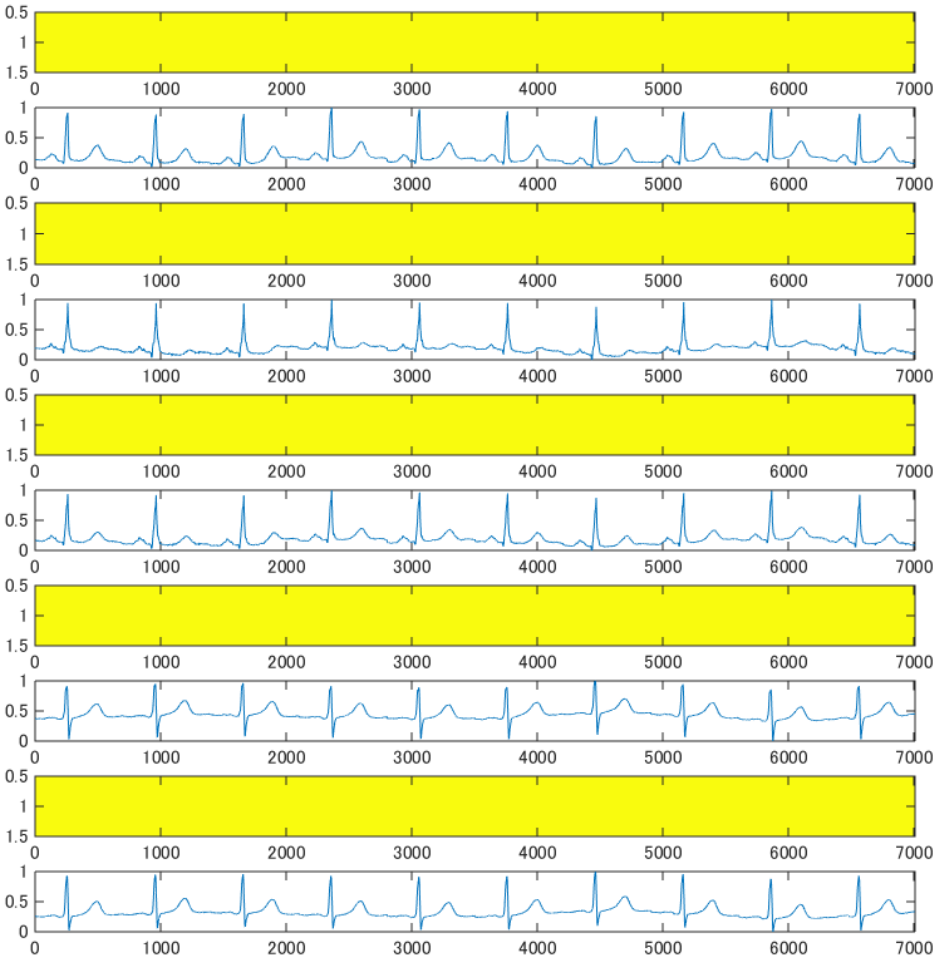


Figure. 3.73: Extraction accuracy of healthy ECG data by proposed discretization method

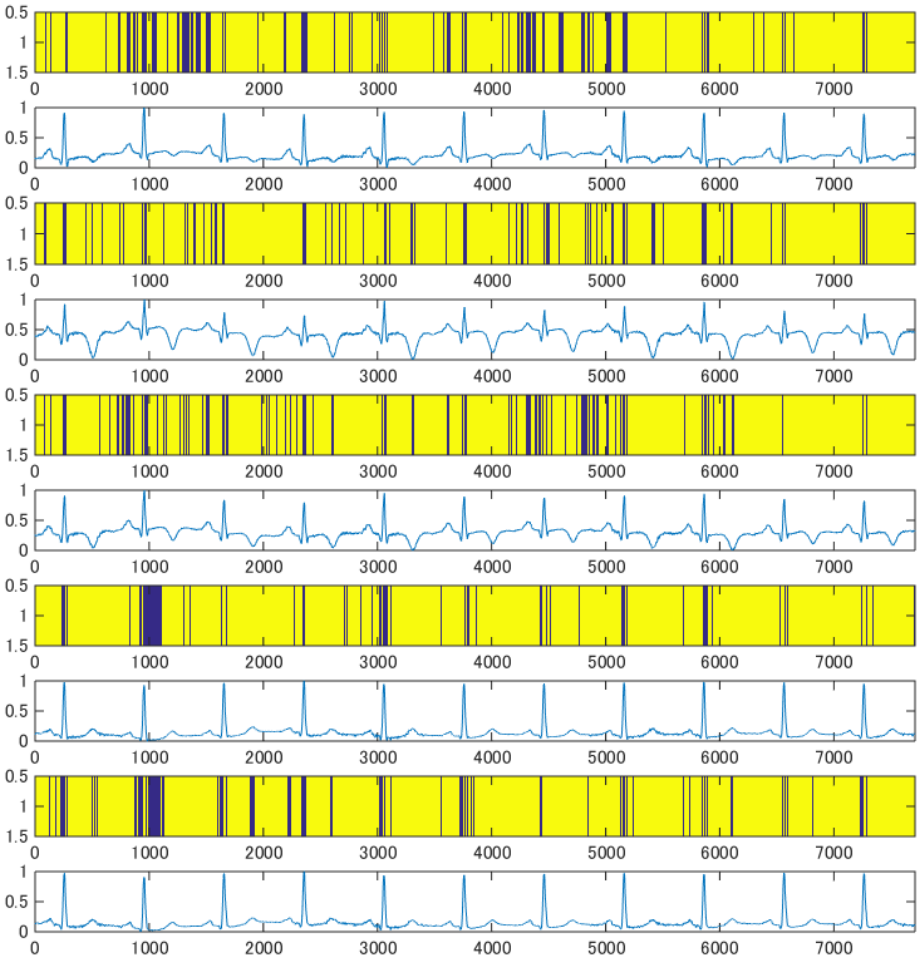


Figure. 3.74: Extraction accuracy of proposed discretization method (θ =15)

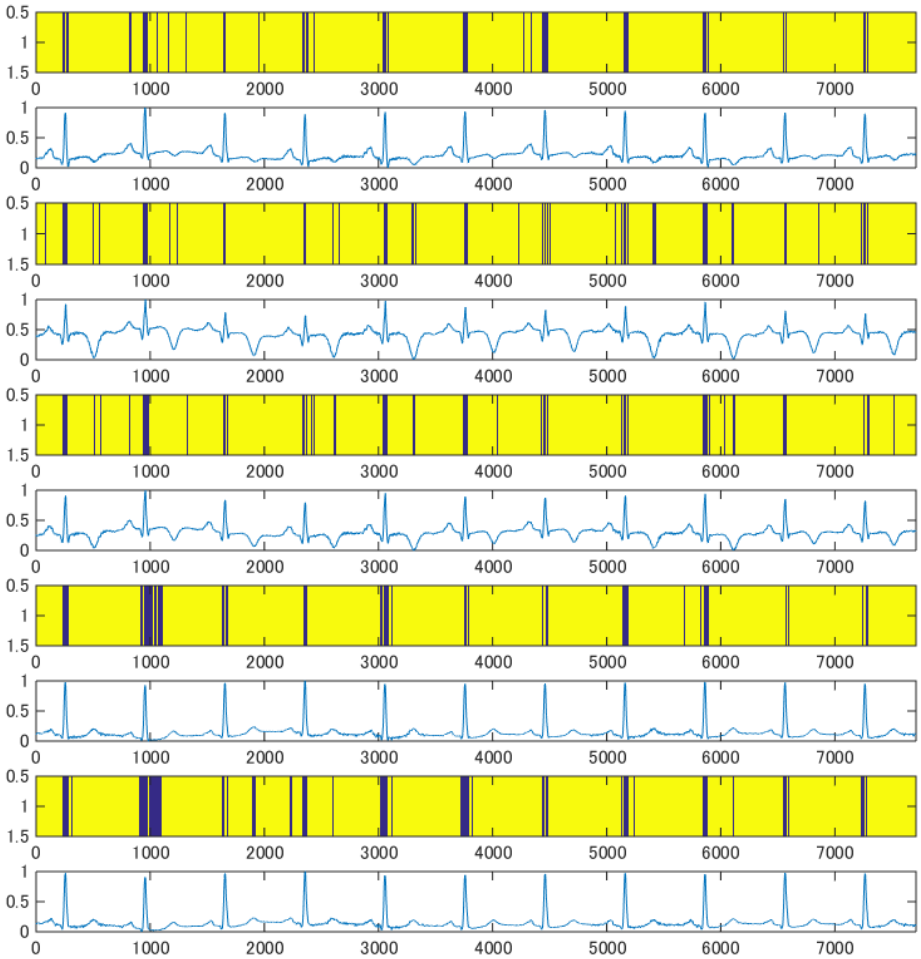


Figure. 3.75: Extraction accuracy of proposed discretization method (θ =20)

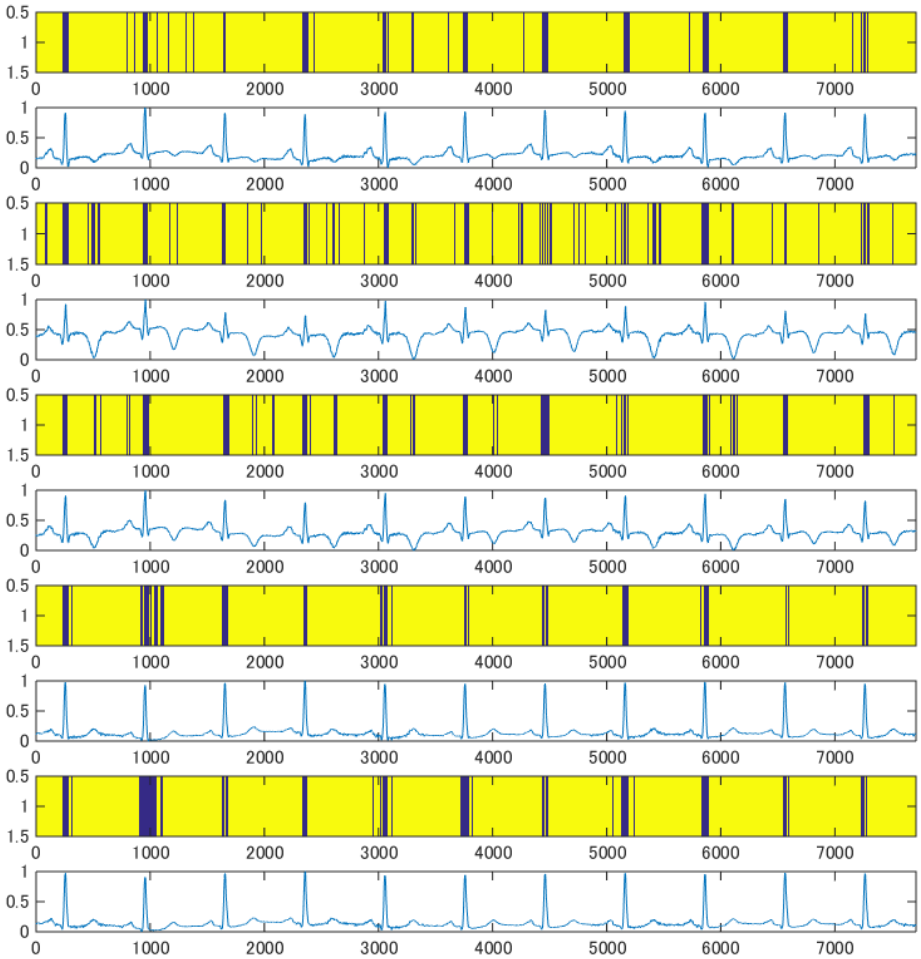


Figure. 3.76: Extraction accuracy of proposed discretization method (θ =25)

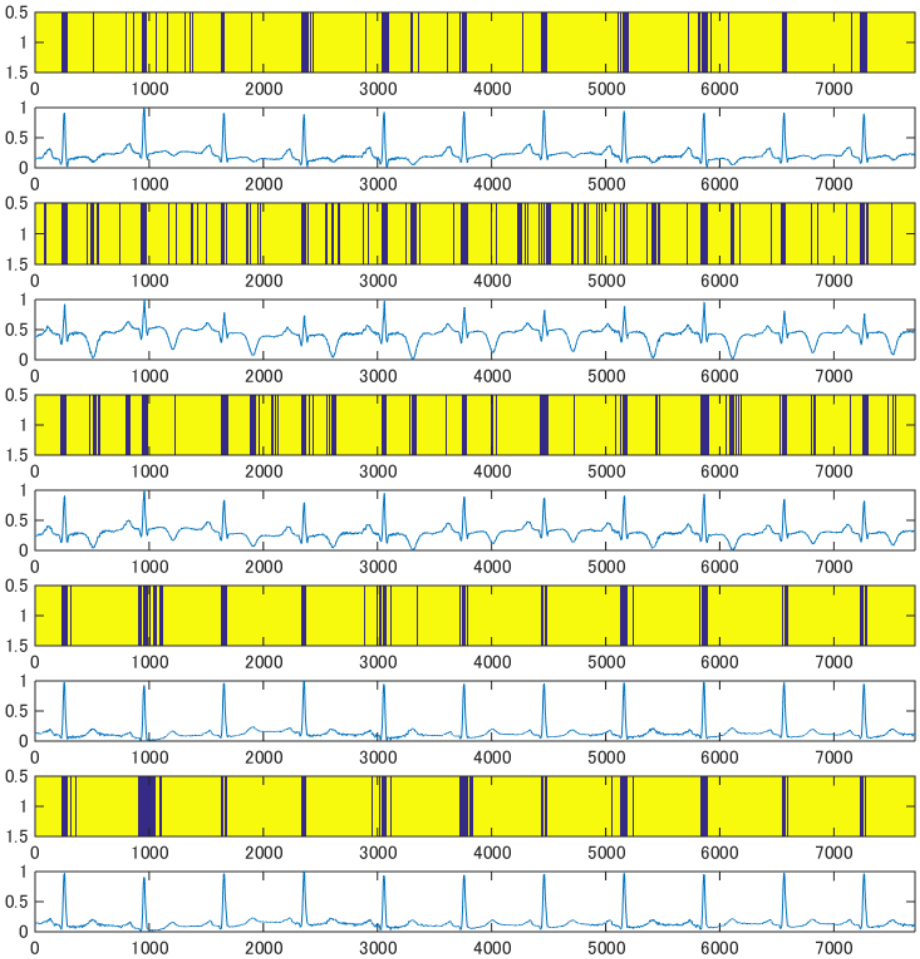


Figure. 3.77: Extraction accuracy of proposed discretization method (θ =30)

3.5 Conclusion

In this section, the previous and proposed discretization method was applied to real ECG (electrocardiogram) data, and the performance is evaluated. In this experiment, a discretization method based on data distribution is newly incorporated into the proposed method in order to deal with the peak in ECG data. As a result, it is shown that the proposed method can extract abnormal waveform that are composed of waves crucial for diagnosis of heart disease. This suggests that the proposed method is available as a new abnormality detector for ECG data.

In the future, we will apply the method to another disease ECG data is made from abnormal waveform and the performance will be evaluated.

References

[1] R. Agrawal and R. Srikant, “Mining sequential patterns,” in 1995 Proc. 11th Int. Conf. on Data Engineering, pp. 314.

[2] F. Takchungm, “A review on time series data mining,” Engineering Applications of Artificial Intelligence, vol. 24, no. 1, 2011, pp. 164181.

[3] Q. Zhao and S. S. Bhowmick, “Sequential pattern mining: A survey. technical report.” CAIS, Nanyang Technological University, Singapore, No. 2003118, 2003.

[4] C. I. Ezeife and Y. Lu, “Mining web log sequential patterns with position coded pre-order linked WAP-tree,” Data Mining and Knowledge Discovery, vol. 10, pp. 538, 2005c Springer Science, Business Media. Inc. Manufactured in The Netherlands.

[5] X. Wu, Y. Wu, Y. Wang, and Y. Li, “Privacy-aware market basket data set generation: A feasible approach for inverse frequent set mining,” in 2005 Proc. 5th SLAM Int. Conf. on Data Mining, pp 103114

[6] A. D. Lattner, A. Miene, U. Visser, and O. Herzog, “Sequential pattern mining for situation and behavior prediction in simulated robotic soccer,” RoboCup 2005: Robot Soccer World Cup IX Lecture Notes in Computer Science, vol. 4020, pp 118129, 2006.

[7] R. Sarno, R. D. Dewandono, T. Ahmad, M. F. Naufal, and F. Sinaga, “Hybrid association rule learning and process mining for fraud detection,” IAENG International Journal of Computer Science, vol.42, no. 2, pp. 5972, 2015.

[8] M. Karaca, M. Bilgen, A. N. Onus, A. G. Ince, and S. Y. Elmasulu, “Exact tandem repeats analyzer (E-TRA): A new program for DNA sequence mining,” J. Genet., vol. 84, pp. 4954, 2005.

[9] H. Ohtani, T. Kida, T. Uno, and H. Arimura, “Efficient Serial Episode Mining with Minimal Occurrences,” in 3rd Int. Conf. on Ubiquitous Information Management and Communication, 2009.

[10] P. Wen-Chi and L. Zhung-Xun, “Mining sequential patterns across multiple sequence databases,” Data and Knowledge Engineering, vol. 68, no. 10, pp. 10141033, 2009.

[11] C. Gong, W. Xindong, and Z. Xingquan, “Mining sequential patterns across time sequences,” New Generation Computing, vol. 26, pp. 7596, 2008.

[12] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.- C. Hsu, “PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth.” in 2001 Proc. 17th Int. Conf. on Data Engineering, pp. 215224.

[13] H. Mannila, H. Toivonen, and A. I. Verkamo, “Discovery of frequent episodes in event sequences,” Data Mining and Knowledge Discovery, vol. 1, pp. 259289, 1997.

[14] Y. Sakurai, C. Faloutsos, and M. Yamamuro, “Stream monitoring under the time warping distance” in 2007 Proc. ICDE, pp. 10461055.

[15] Y. Sakurai, S. Papadimitriou, and C. Faloutsos, “BRAID: Stream mining through group lag correlations.” in 2005 Proc. ACM SIGMOD Conf., pp. 599610.

[16] Y. Zhu and D. Shasha, “StatStream: Statistical monitoring of thousands of data streams in real time” in 2002 Proc. of VLDB, pp. 358369.

[17] T. Miura and Y. Okada, “Detection of linkage patterns repeating across multiple sequential data,” Int. J. Computer Applications, vol. 63, no. 3, pp. 1417, 2013.

[18] N. Miyoshi, T. Shigezumi, R. Uehara, and O. Watanabe, “Scale free interval graphs,” Theoretical Computer Science, vol. 410, no. 45, pp. 45884600, 2009.

[19] N. Korte and R. H. Mohring, “An incremental linear-time algorithm for recognizing interval graphs,” SIAM J. Computing, vol. 18, pp. 6881, 1979.

[20] G. S. Lueker and K. S. Booth, “A linear time algorithm for deciding interval graph isomorphism.” J. ACM, vol. 26, pp. 183195, 1979.

[21] T. Uno, M. Kiyomi, and H. Arimura, “LCM ver.3: Collaboration of array, bitmap and prefix tree for frequent itemset mining,” in 2005 Proc. of 1st Int. Workshop on Open Source Data Mining, pp. 7786.

[22] Y. Liu, M. Kim, J. Takada, T. Kishi, J. Suzuki, and S. Suzuki, “Contact-less Heartbeat Detection Method by using Microwave,” IEICE, pp. 1-6, 2012.

[23] Y. Hirao, J. Yu, Y. Takeuchi, and M. Imai, “Arrhythmia Detection based on deformable model of ECG,” IEICE, pp. 25-30, 2016.

[24] M. Tsuyunashi, K. Oguri, H. Matsuo, and A. Iwata, “The Holter ECG waveform classification using clustering,” IEICE, pp. 1-6, 2003.

[25] D. Obeid, G. Issa, S. Sadek, G. Zaharia and G. El Zein, "Low Power Microwave System for Heartbeat Rate Detection at 2.4, 5.8, 10 and 16 GHz," 1st Int. Symp. Applied Sciences on Biomedical and Communication Technologies, pp 1-2, 2008.

[26] S. Osowski, L.T. Hoai, and T. Markiewicz, "Support vector machine-based expert system for reliable heartbeat recognition," IEEE Trans. Biomed. Eng., vol.51, no.4, pp. 582-589, 2004

[27] J.J. Oresko, Z. Jin, J. Cheng, S. Huang, Y. Sun, H. Duschl, and A.C. Cheng, "A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing," IEEE Trans. Inf. Technol. Biomed., vol.14, no.3, pp. 734-740, 2010.

[28] S. Osowski and T.H. Linh, "Ecg beat recognition using fuzzy hybrid neural network," IEEE Trans. Biomed. Eng., vol.48, no.11, pp. 1265-1271, 2001

[29] http://medical-dictionary.thefreedictionary.com

Acknowledgment

本研究を進めるにあたり，多大なるご指導いただきました岡田吉史准教授に心より感謝申し上げます．岡田先生のお言葉を励みに，本日まで研究を続けることができました．また，お世話になりました板倉賢一教授、工藤康生教授にもお礼申し上げます．最後に，精神的な支えとなってくださった研究室の後輩の皆さん，友人たち，家族に感謝します．