# Cache Learning Method for Terrific Detection of Atrial Fibrillation




Abdul Hakim Mohamed
A'Sharqiyah University



Fadi Abdel Muniem Abdel Fattah
A'Sharqiyah University

# Cache Learning Method for Terrific Detection of Atrial Fibrillation


Mohamed Ezzeldin A. Bashir[1[0000-0002-2292-2677]], Abdul Hakim H.M. Mohamed[1[0000-0001-5879-5385]], Akbar Khanan[1[0000-0002-2226-0001]], Fadi Abdel Muniem Abdel Fattah[1[0000-0002-4665-4777]], Ling Wang[2[0000-0001-8956-0511]], and Keun Ho Ryu[3[0000-0003-0394-9054 ]*]

[1]College of Business Administration, Al-Sharqiyah University, Oman

[2]Department of Computer Science and Technology, School of Computer Science, Northeast Electric Power University, Jilin City, China

[3]Department of Computer Science, College of Electrical and Computer Engineering, Chungbuk National University, Cheongju 28644, Korea

[1][mohamed.bashir, abdulhakim.mohamed, akbar.khanan, fadi.fattah]@asu.edu.om

[2]smile2867ling@neepu.edu.cn

[3]khryu@chungbuk.ac.kr



**Abstract.** Atrial Fibrillation AF reported as the most occurring heart arrhythmia. Steadfast detection of AF in ECG monitoring systems is considerable for early treatment and health risks reduction. Various ECG mining and analysis efforts have addressed a wide variety of technical issues. However, the morphological descriptors are changing along the time within the different patients. As a result, the classification model constructed using old training data is not accurate enough to detect AF. This paper presents an outstanding dynamic learning method to achieve better AF arrhythmia detection in real-time applications. The performance of our proposed technique showed 96.2%, 99.7%, and 99.4% for sensitivity, specificity, and overall accuracy respectively. Accordingly, the proposed Cache learning method can be introduced to improve the performance of the AF intelligent detection systems.



**Keywords:** *Atrial Fibrillation, Cardiac Monitoring, Electrocardiogram (ECG), and* Dynamic Learning.


## 1 Introduction

Among all cardiac arrhythmias, Atrial Fibrillation (AF) reported as the most common one. AF causes the heart to beat in irregular basis, which causes pushing blood ineffectively then the patient suffers from the lack of oxygen. Usually, it affects old peoples over the seventies [1]. AF can upgrade the stroke risk percentage up to 15–20% [2]. The patient with AF, reporting abnormal atrial activation in Electrocardiogram (ECG), due to the random electrical signals generated by sinus node that reflects in continues circulating waves [3]. The ventricular response accordingly in the wrong manner because of the numerous fibrillatory waves circulate unevenly across the atrial myocardium. Hence, there is a miss synchronization between atrial rhythm and


*Corresponding author

ventricular rhythm [4]. Electrocardiogram (ECG) provides an essential tool to detect AF. There are some methods developed to enumerate and detect AF regarding different features related to the ECG parameters [5]. But there are several limitations like the outsized morphological disparity of heart rhythm between different people and in the same person in different situations like sleeping or running or in a silent situation. As a rule, the development of a method with a training set of data after a while, it is necessary to update or moderate to be familiar with the new heart rhythm with different features related to the ECG parameters. In this paper, we propose the Cache learning method to enhance the development of a real-time AF detecting system, concerning the contest for training the classifiers model with a fresh dataset. This paper presents a summary of the related work, the proposed technique, experimental works, and the conclusion.

## 2 Related Work

There are two main philosophies to train the classifier model, the first approach based on splitting the dataset into two main groups with different percentages. One of them usually uses to train the classifier during the training time and another group uses to make sure that the classifier is working effectively by testing it and measuring sensitivity, specificity, and accuracy with an unobserved dataset offered in another group of data. Usually [6]. This approach suffers from so many issues since it is static. Then, the challenge of the morphologies ECG waveforms that differ from time to time with the patient and from different patients concerning sex, age, etc. are not considered. Accordingly, the performance of the classifier looks accepted when using the same patient's dataset for learning it, while its performance often drops very much when the testing process is done through a group of patients. Since the dataset used for learning vary from one patient to another. The literature shows that researches and merchants attempt to concord the mentioned problem in this approach by increasing the amount of dataset that uses for training as much as they can to overcome this problem. This approach deprecated in different parts. First, using a very big training dataset to learn the classifier reflects in the complexity of the system in terms of development and troubleshooting. From a technical point of view, continues learning with up ordinariness morphological reflected in the ECG during the patient surveillance, is something very difficult. Consequently, there is a great chance to miss the detection of specific heart arrhythmia when stratifying the outmoded approach. Additionally, it is unbearable to familiarize the classifier with all ECG waveforms from all anticipated patients [7]. The second approach basses on developing a classifier model to monitor a specific patient [8]. The private learning approach relief the miss matching of the ECG morphological in the learning process with that one in the testing process, it criticized in different aspects. First, splitting down the database according to each patient is very costly especially in a huge database. Furthermore, it is very rare to find patients to involve them in the process of development especially they are sick, and it is very hard for them to spend time in labs. Y. H. Hu et al. [9] introduced a technique that makes the training process self-adapting by utilizing the mixture-of-expert (MOE) to obtain patient readjustment. The self-adapting training method saves the efforts of dividing the database manually. Nevertheless, there are several downsides such as weakness of the sensitivity that generated due to the comparison between different modules, not only that but also it is very expansive because the system depends on developing a local expert for each patient need to be monitor. Furthermore, using more than one classifier at the same time maximize the chance of mistakes. In previous works, M. E. A. Bashir et al. [8] proposed a nested ensemble method to relief the self-adapting technique's problem by adapting the training dataset in a dynamic manner using fresh data, not only that but also acclimatizing the ECG morphological features related to each heart arrhythmia to improve the accuracy. Even though the performance is outstanding but harmonizing the dynamic training side with customized the ECG feature side is very expensive.

## 3 Cache Learning Method

Cache technique attempts to provide a very smart dynamic learning process by offering a renewed training set of data to overcome changes of ECG morphological features, which keep changing through time. The method starts with an initial training set then there will be limited modification to the initial group of the training set when the accuracy dropped down. The current training set will replace incompletely by a new one when there is a limited retreat and the classifier accuracy deteriorated. the new group of datasets that contain current features needed, will introduce to upgrade the current learning dataset. In the worst scenario, all the current data will be replaced with a new set to start a new round with up to date data when there is a significant number of moderations expected. To avoid delay, the technique is based upon in-between files to contain special kinds of data in a smart arrangement technique called Cache, so the change always takes place through the Cache. The Cache technique has four steps, as shown in (Fig.1.) The preliminary learning phase screening, development, and the elimination phase. The positive side and the contribution of this work are reducing the number of times uses to check the main database to the minimum level. Hence improve the accuracy without increasing the computational cost.

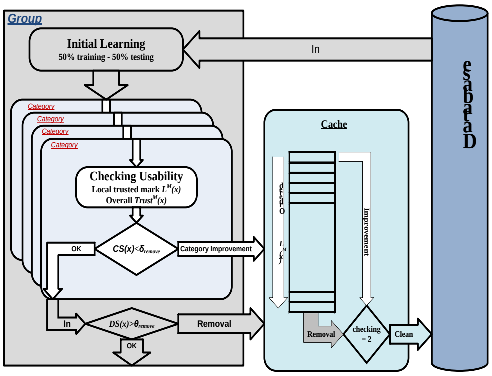


**Fig.1.** Cache learning technique

### 3.1 Preliminary Learning

Preliminary learning is the starting point, the learning process begins with half of the overall data. Choosing the data for this stage happens randomly, there is no factor involves in selecting a specific record (category). So, all records have the same chance to be selected to design the initial training dataset. In this step, there is no kind of check or measurement, also there is not any kind of improvement.

### 3.2 Screening

The screening step is the second one that follows the preliminary stage. $Trust^M(x)$ is the overall trust index that indicates the usability of the training set (x) according to the label assigned to the different types of arrhythmia. The measurement of the $Trust^M(x)$ happens arbitrarily. The $Trust^M(x)$ elaborated by $L^M(x)$ the local trust index. The local trust index $L^M$ (x), generated accumulatively when classifying specific arrhythmia accurately using a specific set of ECG features. The $L^M$ (x) the local trust index increases rapidly by one unit every time the list of features nominated to detect specific arrhythmia succeeds to detect arrhythmia. Formula (1) details the calculation of the $L^M$ (x) local trusted index.

$$L^M(x) = \sum_{f \in features} \beta_f(F,i).C^S(x) \tag{1}$$

The weight of the exact feature to detect specific arrhythmia is represented by (F), (f) represents an index of a specific feature, and the weight of the overall class of features $C^S(x)$ that succeed to detect arrhythmia is given by formula (2).

$$C^S(x) = \sum_{f \in features} \beta_f(F,i) \tag{2}$$

The class of features (F) uses to detect specific arrhythmia (i) is passed to the Function $\beta_f$ (F, i). If the class of the features (F) succeed to detect the Arrhythmia (i) the function returns (+1) and if it fails to detect arrhythmia it returns (-1).

$$\beta_f(F,i) = \begin{cases} +1 & \text{if label}(x) \ = i \\ -1 & \text{otherwise} \end{cases} \tag{3}$$

The overall trust index $Trust^M(x)$ obtained by utilizing $L^M(x)$ the local trust index in formula (4) and the sigmoid function ($0.5 < Trust^M(X) < 1$) that calculate in formula (5).

$$Trust^M(X) = sigmoid \sum_{x=1}^{n} L^M(x) \tag{4}$$

$$sigmoid(x) = \frac{1}{1+exp(x)} \tag{5}$$

### 3.3 Development

The screening step conclude either by the current training set is valid to detect arrhythmias accurately, or there is miss detection with different percentages. Consequently, the dataset that performs oddly is going to be modified either partially or completely by a new fresh training dataset. the development phase has two sub-steps. The first one is labeling the impractical features and or class of features then, substitute it/them with a new selected feature or class. The labeling phase is conducted in the current training set utilized to detect arrhythmias. The current training set is grouped into a set of classes the class is going to be removed if its overall trust score is below than threshold $\delta_{remove}$. Formula 6 explain the calculation of class score:

$$\begin{gathered}\text{if } C^S(x) < \delta_{remove} \\ \text{then remove}\end{gathered} \tag{6}$$

The removed class will be copied in a simple design file called Cache. It will not send back to the main database. The cache arranges according to the class score of the class, Cache allocates the classes in ascending way. So, the class with the smallest $C^S$ will be at the bottom of the file and the opposite is true. The position inside the Cache is dynamic which means every time a removed class arrives; the Cache check it is score and compares it against the other class scores then the class takes the right position. The cache file capacity is limited to 50 % of the overall data size to keep the percentage of the training dataset fixed and not more to the half of the overall data. The process refers to the Cache to choose from the top to replace the removed class and if the performance not improved, then finally a new class from the main database fetched accidentally. The chance $P^C$ to choose specific class is relative to the overall trust that explained in formula (4)

$$p^C(x_{selected/removed}) = \frac{C^S(x_{selected/removed})}{\sum_j C^S(x_j)} \tag{7}$$

Calculation of the probability of the two classes the removed ($x_{removed}$) and the selected ($x_{selected}$) are considered to evade selecting a recently removed class. The current training set is updated by the fresh selected class. Consequently, the screening and development continue and repeated always to keep the classifier performance effective. Starting from the second modification stage, the substitutions take place from the Cache, not from the main database. The selection of the substituted category is depending on the $C^S(x)$, so the highest class on the top of the Cache will be nominated. The movement from the Cache is taking place when the class is nominated twice and fails then returns to the cache after it removed from the current learning dataset. Therefore, it replaced by fresh class from the main database by considering a random probability mentioned in formula (7). The screening and development steps cause continuous movement of the classes from the current training set to the Cache and from Cache to the main database. all classes have the same possibility to be selected in the ingoing update process without any consideration of their previous performances.

### 3.4 Elimination

When partial moderation repeated many times without noticeable improvement in the performance of the classifier, then the development process becomes useless. While it is true when there is a limited number of miss detections in some arrhythmias, and there is no need to hit the Cache many times. Hitting Cache rapidly increases the computation cost very clearly. Accordingly, there is a strong need to replace the whole current learning group of data with a new one. The elimination process removes all the classes in the current group of the training set (x) if the defect score $D^S(x)$ is less than the threshold $\theta_{remove}$ calculation of the defect score is in formula (8).

$$\begin{gathered}\text{if } D^S(X) > \theta_{remove} \\ \text{then remove}\end{gathered} \tag{8}$$

Likewise, the elimination process takes place through the cache like the development process expect the first learning process when the group is selected randomly. The whole cache contains will be removed, if it is reactivated and removed from the active learning set two times. Nevertheless, a fresh set of data nominated but not in a random manner. Formula (7) can be utilized for this purpose and restarted from the beginning again. Bear in mind that, the ratio of the training dataset to the validation dataset is not changing during both development and elimination stages.

## 4 Experimental Work

A database developed in the University of California, Irvine [10] is used in the experimental work of this paper. The database introduced by Waikato Environment for Knowledge Analysis (WEKA). It contains 279 features and 452 records [11]. The normal arrhythmia seen in classes from 01 to 15. There are more than fifteen kinds of different arrhythmias like Old Inferior Myocardial Infarction, Left bundle branch block, Right bundle branch block, Sinus bradycardia, Sinus tachycardia, degree AV block, degree AV block, left ventricular hypertrophy, Ventricular Premature Contraction (PVC), Atrial Fibrillation or Flutter is one of them. Besides the normal rhythm. WEKA 3.6.1 environment is used to conduct the experimental works in personal computer intel CORE i3, 1.70 GHz speed. 4.00 GB RAM. The threshold defined for $\delta_{remove}$ to be 1.0 in formula (6) while $\theta_{remove}$ is set to be 0.5. in formula (8).

the cache learning method experimented to detect Atrial Fibrillation (AF). The cases of AF were increased in the total database by duplicating the cases, generating 21 cases of Atrial Fibrillation among the different arrhythmia and normal rhythm. The Cache learning method equipped by the J48 algorithm as a classifier. The sensitivity, specificity, and accuracy are measured to detail the performance of the Cache method. A confusion matrix is used to present the performance of the classifier, where the FN, FP, TN, and TP, represents the false negative, false positive, true negative, and true positive respectively. The performance of the classifier J45 when feeding by Cache learning to diagnose the Atrial Fibrillation (AF) is summarized in Table 1. It illustrates the performance of the Cache learning method with different ECG morphological features.

**Table 1** Performance of Cache Learning using P, QRS, and T ECG parameters

| *parameters* | *TP* | *FN* | *FP* | *TN* | *sensitivity* | *specificity* | *accuracy* |
|---|---|---|---|---|---|---|---|
| QRS only | 17 | 3 | 4 | 225 | 85.5% | 98.3% | 97.2% |
| QRS +P | 18 | 3 | 3 | 225 | 85.7% | 98.7% | 97.5% |
| QRS +P +T | 19 | 1 | 1 | 228 | 96.2% | 99.7% | 99.4% |

The results show that the performance of the cache learning technique improves to some extent with the QRS complex features when added to the P wave features comparing with the performance with features related to the QRS complex alone. While the performance of the classifier improved very clearly when using all features related to the ECG morphological parameters P, QRS, and T. the literature shows that, so many researchers are utilizing the only QRS mainly the R parameter or P to detects AF. There is not any attempt to utilize the other waves or introducing all waves together. The reason is that such an attempt will increase the computational cost [12-13]. Although the detection of the AF very relates to the R and or P waves, the other waves also affected the AF somehow. The result that summarized in Table 2 illustrate the improvement in the performance of the classifier when introducing all features related to all ECG morphological parameters. It is noticeable, the classifier learned by the Cache learning technique is unique especially when introducing all ECG features related to the P, QRS, and T waves.

**Table 2** Performance of Cache method against other techniques to detects AF

| *Author* | *Database* | *Method* | *sensitivity* | *Specificity* | *Accuracy* |
|---|---|---|---|---|---|
| Christov et al. [12] | ECG | P wave | 95.7% | - | 98.8% |
| Fukunami et al. [13] | ECG | P wave | 91% | 76% | - |
| Trigger methods [7] | ECG | QRS +P+T | 95.0% | 99.6% | 99.2% |
| Cache Learning | ECG | QRS +P+T | 96.2% | 99.7% | 99.4% |

## 5 Conclusion

In this paper, the Cache method was introduced to detects Atrial Fibrillation AF arrhythmia very effectively and efficiently. Cache technique attempts to provide a very smart self-motivated learning process by offering rehabilitated training sets of data to overcome variations of ECG morphological features, which keep changing through time. Various approaches have been used to measure the performance of the Cache learning method. The results validate the outstanding performance of the Cache learning techniques, especially when using all ECG parameters. More experiments will conduct in the future to measure the performance of the Cache learning method under noise.

**Acknowledgement**
This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No. 2019K2A9A2A06020672 and No. 2020R1A2B5B02001717).

## References

1. Mohamed Ezzeldin A. Bashir, Lee DG, Piao Minghao, Shon HS, Ryu KH. Crucial ECG Parameters for Precise Atrial Fibrillation Revealing. In the 4th International Conf FITAT 2011, Chanjue, South Korea.
2. U. Rajendra, M. Sankaranarayanan, J. Nayak, C. Xiang, and T. Tamura.: Automatic Identification of cardiac health using modeling techniques: a comparative study. Inf. Science. 178, 457--4582 (2008).
3. J. Rodrigues, A. Goni, and A. Illarramendi.: Real-time classification of ECG on a PDA, Trans. On IT in B.med. IEEE, 23--33 (2005).
4. Mohamed Ezzeldin A. Bashir, Makki Akasha, D. G. Lee, Min Yi, K. H. Ryu, E. J. Bae, M. Cho, and C. Yoo.: Highlighting the Current Issues with Pride Suggestions for Improving the Performance of Real-Time Cardiac Health Monitoring. DEXABilbao, Spain (2010).
5. G. Bortolan, I. Jekova, and I. Christov.: Comparison of four methods for premature ventricular contractions and normal beats clustering. Com. Card. 30, 921--924 (2005).
6. Mohamed Ezzeldin A. Bashir, Lee DG, Li M, Bae JW, Shon HS, Cho MC, et al. Trigger learning and ECG parameter customization for remote cardiac clinical care information system. IEEE Trans Inf Technol Biomed. 2012;16(4):561–71.
7. Mohamed Ezzeldin A. Bashir, Ryu KS, Yun U, Ryu KH. Pro-Detection of Atrial Fibrillation Using a Mixture of Experts.IEICE Transactions on Information and Systems2012;(12):2982–2990.
8. Mohamed Ezzeldin A. Bashir, Makki Akasha, D. G. Lee, Min Yi, K. H. Ryu, E. J. Bae, M. Cho, and C. Yoo.: Nested Ensemble Technique for Excellence Real-Time Cardiac Health Monitoring. BioComp LasVegas USA (2010).
9. I. Christov and G. Bortolan.: Ranking of pattern recognition parameters for premature ventricular contractions classification by neural networks. Phys. Measure,1281--1290(2004).
10. UCI Machine Learning Repository, “http://www.ics.uci.edu/~mlearn/MLRepository.html”.
11. WEKA web site, “http://www.cs.waikato.ac.nz/~ml/weka/index.html”.
12. Christov, I. Bortolan, G. Daskalov, I.: Sequential analysis for automatic detection of atrial fibrillation and flutter, Comput. In Cardiol.293--296(2001).
13. M. Fukunami, T. Yamada, M. Ohmori, K. Kumagai, K. Umemoto, A. Sakai, N. Kondoh, T. Minamino, N. Hoki.: Detection of patients at risk for paroxysmal atrial fibrillation during sinus rhythm by P wave-triggered signal-averaged electrocardiogram. Circulation 83, 162--169(1991).