A Patient-Adaptable ECG Beat Classifier Using a Mixture of Experts Approach

Yu Hen Hu,* *Senior Member, IEEE,* Surekha Palreddy, and Willis J. Tompkins, *Fellow, IEEE*

Abstract—We present a "mixture-of-experts" (MOE) approach to develop customized electrocardigram (ECG) beat classifier in an effort to further improve the performance of ECG processing and to offer individualized health care. A small customized classifier is developed based on brief, patient-specific ECG data. It is then combined with a global classifier, which is tuned to a large ECG database of many patients, to form a MOE classifier structure. Tested with MIT/BIH arrhythmia database, we observe significant performance enhancement using this approach.

Index Terms— ECG beat classification, MIT/BIH database, mixture of experts, neural network, patient adaptation.

I. INTRODUCTION

COMPUTERIZED electrocardiography is now a well-established practice, after several years of significant progress. Many algorithms have been proposed over years for electrocardiogram (ECG) beat detection and classification. In a clinical setting, such as an intensive care unit, it is essential for automated systems to accurately detect and classify electrocardiographic signals on a real-time basis. Since several arrhythmia are potentially dangerous and life threatening, if not detected within a few seconds to a few minutes of its onset, automated electrocardiographic monitoring assumes a challenging role. Several algorithms have been proposed in the literature for detection and classification of ECG beats and reported results, that leave room for improvement. They include signal processing techniques; such as frequency analysis, template matching, and other parameter extraction methods. Artificial neural networks were also employed to exploit their natural ability in pattern-recognition tasks for successful classification of ECG beat [2], [3], [6]–[8], [23]–[25], [28]–[31].

One major problem faced by today's automatic ECG analysis machine is the wild variations in the morphologies of ECG waveforms of different patients and patient groups. An ECG beat classifier which performs well for a given training database often fails miserably when presented with a different patient's ECG waveform. Such an inconsistency in performance is a major hurdle preventing highly reliable, fully automated ECG processing systems to be widely used clinically.

One obvious approach to alleviate this problem is to use as much training data as possible to develop the ECG classifier. This is the approach taken by all the vendors of ECG processing devices: A large in-house ECG database is developed and maintained to test each ECG processing algorithm to be incorporated into the product. However, such an approach suffers several pitfalls.

1) No matter how large this database may be, it is not possible to cover every ECG waveform of all potential patients. Hence, its performance is inherently limited.
2) The complexity of the classifier grows as the size of the training database grows. When a classifier is designed to correctly classify ECG from millions of patients (if it ever becomes possible), it has to take numerous exceptions into account. The result is a complicated classifier which is costly to develop, maintain, and update.
3) It is practically impossible to make the classifier learn to correct errors during normal clinical use. Thus, it may be rendered useless if it fails to recognize a specific type of ECG beats which occurs frequently in certain patient's ECG records.

The answer, we believe, is to allow the classifier to be "patient-adaptable." That is, to let the classification algorithm adaptable to the special characteristics of each patient's ECG records. For example, we may include the training algorithm and the database used to develop the classifier to be delivered to the users, so that the classification algorithm can be fine-tuned to each patient. Unfortunately, this is impractical for several reasons.

- While it is possible to turn over training algorithms and databases to the users in an academic environment, it is unlikely that any commercial ECG machine vendor is willing to risk revealing their proprietary information to their competitors. Moreover, in-house database often contains millions of ECG records which could be costly to distribute.
- Users often do not want to be bothered by implementation details of an ECG algorithm. Thus, few users will be able to take advantage of this patient-adaptation feature even if it is available.
- Even if a user is willing to perform the patient customization, he or she still have to provide sufficient number of patient-specific training data in order to perform patient-adaptation. Manually editing ECG record is a time consuming, labor intensive task. Hence, the size of

patient-specific training data must be tightly controlled.

In this study, we propose a novel approach to patient-adaptation while avoiding these difficulties: 1) We do not require the factory-trained ECG classifier to provide training algorithms or training databases. Instead, all we need is that this classifier gives both its classification results, as well as an estimate of posterior probability of the feature vector as is drawn from each particular class. Hence, no company proprietary information is needed. 2) A patient-specific classifier will be developed using an automated procedure, without human supervision. 3) Only a brief manually edited patient ECG record (2–5 min) is needed to achieve significant performance improvement.

This proposed approach is based on three popular artificial neural network (ANN)-related algorithms, namely, the self-organizing maps (SOM), learning vector quantization (LVQ) algorithms, along with the mixture-of-experts (MOE) method. SOM and LVQ together are used to train the patient-specific classifier, and MOE is a paradigm which facilitates the combination of the two classifiers (original and patient-specific) to realize patient-adaptation. In MOE, the two classifiers are modeled as two experts on ECG beat classification. The original classifier, called the *Global expert* (GE) in this work, knows how to classify ECG beats for many other patients whose ECG records are part of the in-house, large ECG database. The patient-specific classifier, called the *local expert* (LE) in this work, is trained specifically with the ECG record of the patient. A gating function, based on the feature vector presented, dynamically weights the classification results of the GE's and the LE's to reach a combined decision. The process is analogous to two human experts arriving at a consensus based on their own expertise.

Section II reports the results of literature survey and Section III discusses data acquisition with preprocessing. Section IV discusses the proposed algorithms and the development of experts. Section V reports the results of the classifier on the database records and discusses the results. Section VI is a summary of the findings of this paper.

II. PRELIMINARIES

A. ECG Beat Classification Techniques

Automated ECG beat classification was traditionally performed using a decision-tree-like approach, based on various features extracted from an ECG beat [1], [4], [5], [13], [20], [22]. The features used include the width and height of QRS complex, RR interval, QRS complex area, etc. One of the difficulties is that these features are susceptible to variations of ECG beat morphology and temporal characteristics. As such, the classification rate reported in these earlier efforts are rather moderate.

Artificial neural networks (ANN's) have been widely accepted for pattern recognition tasks. Their abilities to learn from examples and extract the statistical properties of the examples presented during the training sessions, make them an ideal choice for an automated process that imitates human

the purpose of ECG beat detection and classification. Previous reported efforts include [2], [3], [6]–[8], [23]–[25], and [28]–[31].

Hu *et al.* [7] reported the development of an adaptive multilayer perceptron (MLP) for classification of ECG beats. They have achieved an average recognition accuracy of 90% in classifying the beats into two groups; normal and abnormal. In an attempt to classify the beats into 13 groups according to the MIT Database annotations, they have reported an average recognition accuracy rate of 65%. An hierarchical system of the MLP networks which first classify the beat into normal or abnormal, and then classify it into the specific beat type, is developed, which improved the recognition accuracy to 84.5%.

B. Self-Organization Map (SOM) and Learning Vector Quantization (LVQ)

SOM and LVQ are both clustering based algorithms proposed by Kohonen [14], [15]. SOM is an unsupervised on-line clustering technique. In SOM, each cluster center (prototype or code word) is represented by the weights of a neuron which is assigned to a coordinate in the feature map. The SOM training algorithm forces adjacent neurons in the feature map to respond to similar feature (input) vectors. In a way, this feature map is analogous to the spatial organization of sensory processing areas in the brain. Let $m_i(t)$ be denoted as the weights (code word) or the ith neuron in SOM during the time instant t, the weights of SOM then are updated according to the following simple formula:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \qquad (1)$$

$h_{ci}(t)$ is the so-called *neighborhood kernel*, which determine the size of neighborhood of the ith neuron within which all neighboring neurons will be updated in response to the present feature vector $x(t)$. Initially, the neighborhood is large. The size reduces as clustering converges, until no neighboring neurons will get updated.

LVQ is a supervised, clustering-based classification technique which classifies a feature vector $x(t)$ according to the label of the cluster prototype (code word) into which $x(t)$ is clustered. Classification error occurs when the feature vectors within the same cluster (hence, assigned to the same class label) are actually drawn from different classes. To minimize classification error, the LVQ algorithm fine tunes the clustering boundary between clusters of different class labels by modifying the position of the clustering center (prototype or code word). This method is called "learning *vector quantization*" because this clustering based classification method is similar to the "*vector quantization*" method used for signal compression in the areas of communication and signal processing.

According to Kohonen, there are three different LVQ algorithms, called LVQ1, LVQ2, and LVQ3 developed at subsequent stages to handle classification problems with different natures. In this study, the optimized learning-rate LVQ1 and LVQ3 algorithms were used for the training and fine-tuning of the code book respectively. In LVQ1, for a given input vector $x(t)$, a code word y_c is found such that

The *code word* is then updated as follows:

$$y_c(t+1) = [(1 - s(t)\alpha_c(t)]y_c(t) + s(t)\alpha_c(t)x(t) \quad (3)$$

where $s(t) = +1$ if the classification is correct [i.e., $x(t)$ and y_c have the same class label] and $s(t) = -1$, otherwise. $\alpha_c(t)$ is a time-varying learning rate. Other code words in the code book remain unchanged. LVQ3 differs from LVQ1 in how the code words are updated: Assuming that $x(t)$ falls within a *window* between two adjacent clusters with corresponding code words y_i and y_j. Suppose that x and y_j belong to the same class, and x and y_i belong to different classes, then both these code words will be updated in LVQ3:

$$y_i(t+1) = y_i(t) - \alpha(t)[x(t) - y_i(t)] \quad (4a)$$
$$y_j(t+1) = y_j(t) + \alpha(t)[x(t) - y_j(t)]. \quad (4b)$$

On the other hand, if both y_i and y_j belong to the same class as $x(t)$, and $x(t)$ fall in a window centered at the cluster boundary of these two classes, then

$$y_k(t+1) = y_k(t) + \varepsilon\alpha(t)[x(t) - y_k(t)],$$
$$k \in \{i, j\}, 0.1 < \varepsilon < 0.5. \quad (5)$$

The optimal value of ε depends on the size of the window, being smaller for narrower windows. This algorithm is self-stabilizing, and optimal placement of the y_i does not change in continual training.

Software packages of both SOM and LVQ are available in the public domain,[1] and the application of these packages to the ECG beat classification problem is straight forward. The adaptation parameters in these packages (SOM_PAK and LVQ_PAK) were carefully fine tuned while developing the classifiers. As such, the development of the code book and eventually decision boundary can be made completely transparent to the user. Moreover, performance obtained using these package is very competitive compared to other approaches. In this research work, we first apply SOM to a set of training feature vectors. The resulting code book (prototypes) then will be submitted to the LVQ_PAK to facilitate fine tuning and classification.

III. MIXTURE OF EXPERTS (MOE)

This user adaptation problem bears certain resemblance to the incremental learning problem in that new data are to be incorporated to improve existing classifier's performance. However, the black-box model of the existing classifier prevents us from directly modifying the classifier structure as incremental learning algorithms do. Instead, we propose a different method called the MOE, to circumvent this problem.

The MOE approach was proposed by Jacobs *et al.* [9]–[12], [16], [26], [27]. The basic notion is that linear combinations of several statistical estimates can perform better than any individual estimate. This strategy is not new. It is a well known fact that a panel of experts often arrive at a better diagnosis than any single expert, because each expert is able to contribute from his/her own expertise.

The basic idea is to leave the existing black-box classifier intact. Instead, we use the given small, user-specific training data set to develop a LE classifier. Then we invoke a modified MOE approach to combine these two classifiers, hoping to achieve better performance.

To apply the MOE approach to solve the customization problem, we employ two experts: a GE and a LE. The GE represents the ECG beat classifier developed in factory. Thus, it is trained to classify all types of ECG beats present in the in-house ECG database. The LE represents a specialized ECG beat classifier, trained on a small segment of annotated ECG beats taken from the specific patient. As such, the GE and the LE are endowed with complementary knowledge bases, and can work together to reach a better decision than any one can reach individually.

The expert network is a combination of the GE and LE classifiers. Let $y_1(x)$ and $y_2(x)$ be the output (row) vectors of the two respective GE and LE classifiers. Each element of each vector indicates the degree of proximity of an unknown ECG beat to a predefined ECG beat class (category). In the MOE method, the combined ith output vector of both the experts is given by

$$h_{ic}(x) = g_{1c}(x)y_{1c}(x) + g_{2c}(x)y_{2c}(x) \quad (6)$$

where x is the input feature vector, g_{jc}, $j = 1, 2$ are the weighting vectors for each expert from a *gating network* and are defined by

$$g_{jc}(x) = \frac{\exp(v_{jc}^t x)}{\sum_{j=1}^{2} \exp(v_{jc}^t x)} \quad (7)$$

where v_{ij}'s are the weight vectors of the gating network. Note that $\sum_{j=1}^{2} g_{jc}(x) = 1$.

Theorem 1: Define $R[y_i(x)] = \{x | x \in X, \text{ and } y_i(x) = z(x)\}$, $i = 1, 2$ to be the subregion in the feature space where the classifier $y_i(x)$ makes correct classification of x and let $R[y(x)]$ be defined the same way. Assume $y_i(x) \in \{0, 1\}$ and $z(x) \in \{0, 1\}$, then

$$R[y(x)] \subseteq R[y_1(x)] \cup R[y_2(x)]. \quad (8)$$

Proof: We need only to prove that if both $y_1(x)$ and $y_2(x)$ misclassify a given feature vector x, then $y(x)$ cannot give correct classification on x. Since the correct classification output $z(x)$, the combined output $y(x)$, and individual classifier output $y_1(x)$ and $y_2(x)$ are all binary vectors of the same dimension, if both classifiers misclassify a given feature vector x which belongs to class c, we must have, for the cth elements of these binary vectors

$$z_c(x) \oplus y_{1c}(x) = z_c(x) \oplus y_{2c}(x)$$
$$= 0$$

where "$\oplus$" is the "exclusive-OR" operator in Boolean algebra. Since from (7), $g_{1c}(x) + g_{2c}(x) = 1$, we conclude $y_c(x) = 0$, if $y_{1c}(x) = y_{2c}(x) = 0$, and $y_c(x) = 1$, if $y_{1c}(x) = y_{2c}(x) = 1$. Hence, $z_c(x) \oplus y_c(x) = 0$. In other words, $y(x)$ must also

of $g_{1c}(x)$ and $g_{2c}(x)$. This is to say, if $x \in R[y_1(x)]$, and if $x \in R[y_2(x)]$, then $x \in R[y(x)]$. $\blacksquare$

The implication of Theorem 1 is that the maximum performance enhancement of a MOE approach occurs when $R[y_1(x)] \cap R[y_2(x)] = \phi$ (empty set). An example is to designate each classifier to be responsible for classifying a particular class. The assumption that $y_i(x) \in \{0, 1\}$ is essential in this theorem. If $y_i(x) \in [0, 1]$ (interval between zero and one), it is possible to find a counter example. Let $y_1(x) = [0.1\ 0.5\ 0.4]$, $y_2(x) = [0.5\ 0.1\ 0.4]$, and $z(x) = [0\ 0\ 1]$. Then $y(x) = [0.1g_1 + 0.5g_2\ 0.5g_1 + 0.1g_2\ 0.4]$. If $g_1 = g_2 = 0.5$, then $y(x) = [0.3\ 0.3\ 0.4]$ which yields correct classification.

On the other hand, whether $y_i(x)$ takes binary values or not, if both classifiers make correct classification, so will the combined classifier.

Theorem 2: With the same definitions as in Theorem 1, and $y_i(x) \in [0, 1]$

$$R[y_1(x)] \cap R[y_2(x)] \subseteq R[y(x)]. \tag{9}$$

Proof: Assume x [class c^*, and $y_{1c^*}(x) = \max_c y_{1c}(x)$, $y_{2c^*}(x) = \max_c y_{2c}(x)$. Then

$$\begin{aligned} h_{c^*}(x) &= g_1(x)y_{1c^*}(x) + g_2(x)y_{2c^*}(x) \\ &= \max_c \left[g_1(x)y_{1c}(x) + g_2(x)y_{2c}(x) \right]. \end{aligned} \tag{10}$$

Thus, the output $y_{c^*}(x)$ is correctly classified. $\blacksquare$

From Theorem 2, it is clear that if both classifiers #1 and #2 correctly classify a pattern x, then the combined classifier will also correctly classify the same pattern. Hence, this pattern can be excluded from the user-adaptation training set as it will not affect the result.

Adaptation Algorithm: Based on the result indicated in Theorems 1 and 2, the design objective of the MOE network in (3) is to devise a training algorithm to estimate the parameter vectors $\{v_{ic}; i = 1, 2\}$. Given that $y_1(x)$ and $y_2(x)$ are fixed classifiers, this problem can be solved by a gradient procedure as follows: Let us assume $\{x_k; 1 \le k \le K', x_k[X]\}$ be a set of training data used for searching the optimal gating functions $g_{1c}(x)$ and $g_{2c}(x)$, such that the square error at the output $E_c = (1/2K') \sum_{k=1}^{K'} \|z_c(k) - y(v_{1c}, v_{2c}, x_k)\|^2$ is minimized. A gradient search algorithm can be devised as follows:

$$v_{ic}(t+1) = v_{ic}(t) - \mu \nabla_{Vi} E_c. \tag{11}$$

The initial values of v_{1c} and v_{2c} are set to be the centroids of the regions $R[y_1(x)]$ and $R[y_2(x)]$, respectively, for x in the user-specific training data set. The gradient of E_c with respect to v_{ic} can be calculated as

$$\nabla_{Vi} E_c = \frac{1}{2K'} \sum_{k=1}^{K'} [y_c(v_{1c}, v_{2c}, x_k) - z_{kc}] \\ \bullet \nabla_{Vi} y_c(v_{1c}, v_{2c}, x_k) \tag{12}$$

$$= \frac{1}{2K'} \sum_{k=1}^{K'} \left\{ [y_c(x_k) - z_{kc}] \bullet f'[h_c(x_k)] \right. \\ \left. \bullet \left[\sum_{j=1}^{2} y_{jc}(x_k) \bullet \nabla_{vi} g_{jc}(v_{1c}, v_{2c}, x_k) \right] \right\} \tag{13}$$

where $h_c(x) = g_{1c}(x) \bullet y_{1c}(x) + g_{2c}(x) \bullet y_{2c}(x)$. In (13), we assumed the transfer function f is a differentiable threshold function, and is applied to the vector, element by element. Finally, with (13), we have as shown in (14) at the bottom of the page. Hence, for $i, j \in \{1, 2\}, i \ne j$, we have

$$\nabla_{Vi} E_c = \frac{1}{2K'} \sum_{k=1}^{K'} \sum_{c=1}^{C} x_k \{ [y_c(x_k) - z_{kc}] \bullet f'[h_c(x_k)] \} \\ \bullet [y_{ic}(x_k) - y_{jc}(x_k)] \bullet g_{1c}(x_k) \bullet g_{2c}(x_k). \tag{15}$$

Note that in above derivation, the error E is accumulated over the entire epoch (K' feature vectors). The summation over k may be removed if we use on-line update of v_i's for each sample. This yields the following expression for $i, j \in \{1, 2\}, i \ne j$:

$$\nabla_{Vi} E_c = x_k \bullet [y(x_k) - z_k]^t \bullet \text{diag}\{f'[h(x_k)]\} \\ \bullet [y_{ic}(x_k) - y_{jc}(x_k)] \bullet g_{1c}(x_k) \bullet g_{2c}(x_k)\}. \tag{16}$$

Clearly, we have $\nabla_{V1} E_c = -\nabla_{V2} E_c$. This is not surprising with two parameter vectors arriving at a decision hyperplane $(v_{1c} - v_{2c})^t x = 0$.

Until now, we have assumed that the user-specific ECG beat classifier $y_{2c}(x)$ is readily available. However, in reality it needs to be trained with the user-specific training data set. Also, the combined classifier $y_c(x)$ needs to be trained by the same data set in order to determine the gating network parameters. Therefore, if $y_{2c}(x)$ is trained to 100% accuracy

---

$$\nabla_{vi} g_{jc}(v_{1c}, v_{2c}, x_k) = \frac{\nabla_{vi} \exp(v_{jc}^t x_k)}{\sum\limits_{m=1}^{2} \exp(v_{mc}^t x_k)}$$

$$= \frac{\nabla_{vi}[\exp(v_{jc}^t x_k)] \bullet \sum\limits_{m=1}^{2} \exp(v_{mc}^t x_k) - \exp(v_{jc}^t x_k) \bullet \nabla_{vi} \exp(v_{ic}^t x_k)}{\left[\sum\limits_{m=1}^{2} \exp(v_{mc}^t x_k) \right]^2}$$

on the user-specific data set, then the gating network of choice may be $g_{2c(x)} \neq 1$ and $g_{1c}(x) \neq 0$. In light of the results of Theorems 1 and 2, we devised the following strategy to alleviate this problem: First, we construct the user-specific training data set to contain only those feature vectors which the original classifier misclassified. We further partition this training data set into two subsets: one for the training of the user-specific classifier $y_{2c}(x)$, and the other for estimating $g_{1c}(x)$ and $g_{2c}(x)$.

IV. EXPERIMENT

The purpose of this experiment is to demonstrate the usefulness of the proposed user-adaptation procedure. In particular, we will show that an ECG beat classifier trained on general patient records does not perform well when presented with patient records which contain rare beat types. Moreover, we show that the performance of the MOE classifier is able to gain significant performance enhancement with a small amount of annotated patient specific training data.

A. Data Preparation

In this study, we concentrate on the classification of ventricular ectopic beats (VEB's). The 48 records (tapes) from MIT/BIH ECG arrhythmia database [17], [19] are used for the development and evaluation of the classifier. The availability of annotated MIT/BIH database has enabled the evaluation of performance of the proposed beat classification algorithm. The American Association of Medical Instrumentation (AAMI)-recommended practice [18] has provided a protocol for a reproducible test with realistic clinical requirements, emphasizing tape-by-tape presentation of results that estimate an algorithm's ability to detect events of clinical significance.

Accompanying each tape in the MIT/BIH database is an annotation file in which each ECG beat has been identified by expert cardiologist annotators. These labels are referred to as "truth" annotations and are used in training (developing) the classifiers and also to evaluate the performance of the classifiers (experts) in testing phase. According to the AAMI-recommended practice, records containing the paced beats (four records) can be excluded from the reporting requirements. Since this study is to evaluate the performance of a classifier that can identify a premature ventricular contraction (PVC), certain records in the database with no PVC's (11 records) were excluded from the study, leaving 33 records of interest. These excluded records are listed in Table I. Data from channel 1, down-sampled to 180 samples/s were used in this study. The selected files consist of 13 records (numbered from 100–124, inclusive, with some numbers missing) and 20 records (numbered from 200–234, inclusive, with some numbers missing). The first group is intended to serve as a representative sample of a variety of waveforms and artifacts which an arrhythmia detector might encounter in routine clinical use. Records in the second group include complex ventricular, junctional, and supraventricular arrhythmias and conduction abnormalities. Several of these records are expected to present significant difficulty to arrhythmia detectors,

TABLE I
RECORDS OF MIT/BIH DATABASE THAT WERE EXCLUDED FROM THE STUDY

Excluded files from MIT/BIH Arrhythmia Database	
Paced beats	102, 104, 107, 217
No PVCs	101, 103, 112, 113, 115, 117, 122, 212, 220, 222, 232

TABLE II
FOUR CATEGORIES OF INTEREST INTO WHICH THE ECG BEATS OF THIS STUDY ARE CLASSIFIED

	Beat Labels (AAMI recommended practice)
N	any beat that does not fall into the V, F or Q categories (normal beats, supraventricular ectopic beats, and others)
V	a ventricular ectopic beat (VEB): a ventricular premature beat, R-on-T ventricular premature beat, or ventricular escape beat.
F	fusion of ventricular and normal beat
Q	unclassifiable beats

variation, and signal quality. These records were reported to have gained considerable notoriety among database users [18].

In this experiment, we use the first group of files as the training data to develop a GE classifier which is able to classify typical ECG beats. The second group of 20 records is used to simulate the ECG records of 20 patients, which are to be classified by the GE classifier. Since these records consist of less-frequently seen beats, it is expected that the GE classifier will not perform well. If this GE classifier were a commercial device, it will be deemed not-applicable (due to low performance) to many of these 20 test records. However, with the MOE approach, we will adapt this GE classifier with a LE classifier to gain significant performance enhancement at low cost.

The beats in the MIT/BIH database are of several different types. In this study, we are interested in identifying four different categories, as indicated in Table II. Each of the four categories included beats of several types as shown in Table III. The AAMI convention was used to combine the beats into four classes of interest.

B. Training and Testing Procedure

In this study, a GE classifier was developed with SOM and LVQ algorithms using the data from the records of the first group (100–124). Before testing the records, a LE classifier was developed for each of the records in the second group using the first 2.5 min of data. The rest of the record is then tested using the mixture of global and LE's as explained before. Since each record in the MIT/BIH database is of length 30 min, the 2.5 min segment account for 1/12th of total available patient specific data and contains approximately 150 ECG beats. In practice, the attending cardiologist or any expert in ECG beat annotation will have to annotate a brief segment of patient-specific ECG in order to take advantage of the MOE approach. We believe that this is a reasonably small cost compared to the potential gain in performance enhancement. In future, we will explore a more effective method to further

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.

fastcase®
Smarter legal research.