# Feature Selection using a Genetic Algorithm for the Detection of Abnormal ECG Recordings

**H. Altay Güvenir**
**Bilkent University, Department of Computer Engineering**
**Ankara, Turkey**
and
**Burak Acar**
**Stanford University, School of Medicine**
**Stanford, U.S.A.**

## ABSTRACT

A new classification algorithm, called CFI (for *Classification on Feature Intervals*), is developed and applied to problem of detecting abnormal ECG signals. The domain contains records of patients with known diagnosis. Given a training set of such records the CFI learns how to detect arrhythmia. CFI represents a concept in the form of *feature intervals* on each feature dimension separately. Classification in the CFI algorithm is based on a real-valued voting. A genetic algorithm is used to select the set of relevant features. Each selected feature equally participates in the voting process and the class that receives the maximum amount of votes is declared to be the predicted class. The performance of the CFI classifier is evaluated empirically in terms of classification accuracy and running time.

**Keywords:** ECG, Arrhythmia detection, machine learning, feature selection, voting feature intervals.

## 1. INTRODUCTION

Researchers working on artificial intelligence have created many algorithms that successfully learn straightforward abilities. If the context is well-defined and the bounds of the problem can be correctly encoded for the computer, then these algorithms can often pick up a pattern and learn to predict it successfully. Inductive learning is a well-known approach to automatic knowledge acquisition of such patterns and classification knowledge from examples.

In several medical domains the inductive learning systems were actually applied; for example, two classification systems are used in the localization of primary tumor, the prognostics of recurrence of breast cancer, the diagnosis of thyroid diseases, and in rheumatology [13]. The CRLS is a system for learning categorical decision criteria in biomedical domains [19]. The case-based BOLERO system learns both plans and goal states, with the aim of improving the performance of a rule-based system by adapting the rule-based system behavior to the most recent information available about a patient [17]. The DIAGAID is a program that uses the connectionist approach to determine the diagnostic value of clinical data [8].

Classification learning algorithms are composed of two components; namely, the *training* and the *prediction* (*classification*). The training phase, using some induction algorithms, forms a model of the domain from the training examples encoding some previous experiences. The classification phase, on the other hand, uses this model to predict the class that a new instance (case) belongs to.

The main requirement for such a system is to achieve a high prediction accuracy. Furthermore, a classification learning algorithm is expected to have a short training and prediction times. Such a system should be robust to noisy training instances. Also, in some real-world domains, both training and test instances may contain some missing values. Features (attributes) that are used to encode instances may have different levels of relevancy to the domain. A classification learning system should be able to learn and/or incorporate information about the weights of the features. Another requirement might be the comprehensibility of the learned knowledge by human experts. The advantage of this trait is two folded. Firstly, the human experts can check and verify the learned classification knowledge before it is put to use in real-world domains. Secondly, some previously unknown facts and patterns may be brought to the attention of human experts, leading to interesting discoveries in the field.

Previously developed machine learning algorithms usually possess some of these characteristics, yet fail to satisfy the others. For example, some algorithms, (e.g., the nearest neighbor and the instance based learning algorithms [1, 5]) develop a model of the domain quickly, but it may take quite a long time to make a prediction using this model. On the other hand, some algorithms (e.g., the neural networks) can make a fast prediction, however the knowledge they learn is hard to understand and verify for humans.

Success of a classification learning algorithm, in terms of the criteria mentioned above, is directly related to the scheme used for representing the classification knowledge learned. In this paper we present a knowledge representation technique called *classification on feature intervals* (CFI, for short). The representation in CFI is based on *Feature Projections* that has been used previously in CFP [11] and k-NNFP [2]. The CFI, which is a non-incremental and supervised learning algorithm, is applied to the detection

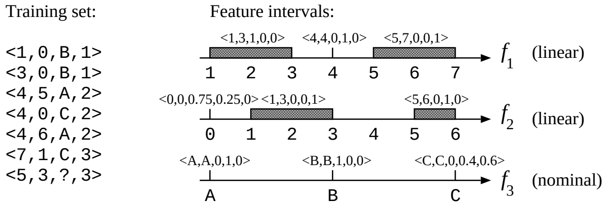


Figure 1: An example training set and the feature intervals constructed by CFI.

of arrhythmia in ECG (electrocardiogram) signals. Here, we show that CFI algorithm results in highly accurate predictions, has short training and classification times, is robust to noisy training instances and missing feature values, can use instances with missing feature values, and produces a human readable model of the classification knowledge.

The rationale behind knowledge representation based on feature intervals is that human experts maintain knowledge in this form, especially in medical domains. The input to CFI training algorithm is a set of training instances that are the descriptions of subjects with known diagnoses. Learning from these training examples, CFI constructs a representation of the classification knowledge inherent in these examples. This knowledge is represented as the projections of the training dataset by *feature intervals* on each feature dimension separately. Then, for each feature dimension, projection points with similar characteristics are grouped into *intervals*. Therefore, an interval represents a set of feature values that yield the same classifications.

When diagnosing a new subject, each feature participates in the voting process and the diagnosis (abnormal or normal) that receives the maximum amount of votes is predicted as the diagnosis of that subject. Since each feature participates independently of the others, both in learning and classification, CFI enables an easy and natural way of handling missing feature values by simply ignoring them. That is, features whose values are unknown do not participate in the voting.

The next section will describe the CFI algorithm in detail. Section 3 describes the genetic algorithm used for feature selection. In Section 4, the problem of arrhythmia detection is explained. Application of the CFI algorithm to this domain is discussed in Section 5. Finally, the last section concludes with some remarks and plans for future work.

## 2. THE CFI ALGORITHM

The CFI classification algorithm is an improved version of the early FIL, VF1 and VFI5 algorithms [3, 7, 10]. Here, the CFI algorithm is described in detail and explained through an example.

### Knowledge Representation

The CFI classification algorithm represents a concept description by a set of feature intervals. The classification of a new instance is based on a voting among the classifications made based on the value of each feature separately. Each training example is represented as a vector of nominal (discrete) or linear (continuous) feature values plus a label that represents its associated class. The CFI algorithm first projects all training instances on each feature separately. Using the projections of the training examples, it constructs a set of intervals for each feature. An interval is either a *range* or a *point* interval. A range interval is a set of consecutive values of a given feature with the same class value, whereas a point interval is defined as a single feature value. For range intervals, lower and upper bounds of the range value, its class value and the vote are maintained. For point intervals, on the other hand, the lower and upper values are the same, but there may be several class values. Therefore, an interval is represented as a vector, whose first two elements store the lower and upper bounds and the remaining elements correspond to the votes for each class, as shown below:

$$< lb, ub, V_1, V_2, \dots V_k > .$$

Here, $k$ is the number of classes in the domain, and $V_i$ represents the vote of the interval for class $C_i$.

An example training data set and the corresponding feature intervals constructed by the CFI algorithms is shown in Figure 1. The example domain consists of three features, namely $f_1$, $f_2$, and $f_3$, the first two of which are linear and the last one is a nominal feature. The nominal feature can take values from the set $\{A, B, C\}$. The class labels are $C_1$, $C_2$, and $C_3$. There are seven training instances in this example.

### Training

The training process in the CFI algorithm is shown in Figure 2. For each feature $f$, first all training instances are sorted with respect to their values for $f$, forming their projections on $f$. A point interval is constructed for each projection. The lower and upper bounds of the interval are equal to the $f$ value of the corresponding training instance. Its vote for the class of the training instance is the reciprocal of the number of times that class occurs in the all training set. This normalization is to eliminate the effects of uneven class distributions in the training set. The votes for the other classes is 0. If the $f$ value of a training instance is unknown (represented by "?" in Figure 1), it is simply ignored for $f$. Then, if there are several point intervals at the same $f$ value, then they are combined into one, by adding the class votes. So that, at the end of point interval construction, there is exactly one point interval for each distinct value of $f$ in the training set. For example, the first interval for $f_2$ in Figure 1 is $\langle 0, 0, 1, 1/3, 0\rangle$. The second and third point intervals are $\langle 1, 1, 0, 0, 1\rangle$, and $\langle 3, 3, 0, 0, 1\rangle$, respectively. Then, only for linear features, CFI tries to generalize the point intervals. Consecutive point intervals whose highest votes are for the same class are merged forming range inter-

```
train(TrainingSet):
begin
    for each feature f
        /* sort TrainingSet with respect to f */
        sort (f, TrainingSet)
        /* construct a list of point intervals using
           feature values and class labels */
        interval_list ← make_intervals (f, TrainingSet)
    if f is linear
        /* join adjacent point intervals
           to form range intervals */
        interval_list ← generalize (interval_list)
    intervals[f] ← normalize_votes (interval_list)
end.

generalize (interval_list)
begin
    I = first interval in interval_list
    while I is not empty do
        I' is the interval after I
        if majority_class(I) = majority_class(I')
        /* majority_class of an interval
           is the class with the highest votes */
        then merge I' into I
        else I ← I'
end.
```

Figure 2: Training in the CFI algorithm.

```
classify(q): /* q: query instance to be classified */
begin
    for each class c /* initialize total votes */
        vote[c] = 0
    for each feature f
        if q_f value is known
            I = search_interval(f, q_f)
            for each class c
                vote[c] = vote[c] + interval_vote(I, c)
    return the class c with the highest vote[c];
end.
```

Figure 3: Classification in the CFI algorithm.

vals. In the example above, the second and third point intervals of $f_2$ are merged into the range interval $\langle 1, 3, 0, 0, 1\rangle$. In the last step of the training process, the votes of each interval are normalized so that the total votes of the interval for all classes is 1. So, following the example in Figure 1, the first interval on $f_2$ becomes $\langle 0, 0, 0.75, 0.25, 0\rangle$.

**Classification**

The classification (querying) process in the CFI algorithm is given in Figure 3. The classification in CFI involves a voting scheme where each feature casts its vote. The process starts by initializing the votes of each class to zero. If the value of the query instance for a feature $f$ is unknown (missing), then that feature does not involve in the voting. That is the features containing missing values are simply ignored. If the $q_f$ value is known, the interval $I$ into which $e_f$ falls is searched. If the $q_f$ value does not fall in any interval on $f$, then again the feature $f$ does not participate in the voting. If an interval $I$ is found that includes the $q_f$ value, then the votes of $I$ are the votes that $f$ casts in the voting. Since the sum of the votes of an interval is normalized to 1, during the training, each feature has an equal power in the voting. Once all the features have completed casting their votes, the class that received the highest amount of votes is predicted to be the class of the query instance.

This implementation of the CFI algorithm is a *categorical classifier*, since it returns a unique class for a query instance [14]. A unique class is predicted for the query instance in order to compare this predicted class with the actual class of the query instance. This enables us to measure the performance of our classifiers according to the most commonly used metric, which is the the percentage of correctly classified query instances over all query instances. On the other hand,

$$\frac{vote[C_j]}{\sum_{i=1}^{k} vote[C_i]}$$

can be used as the probability of class $C_j$ which makes the CFI algorithm a more general classifier. In that case, the CFI algorithm returns a predicted probability distribution over all classes. Although a class is returned as the prediction of the query instance as an output of the CFI classifier , the votes received by each class is also available as an output to the user providing him/her with the level of confidence in the prediction.

Continuing with the example in Figure 1, let the query instance be $\langle 6, ?, C\rangle$. Since the $f_2$ value of the query instance is unknown, the feature $f_2$ does not participate in the voting. The votes of $f_1$ and $f_3$ are $\langle 0, 0, 1\rangle$ and $\langle 0, 0.4, 0.6\rangle$, respectively. The total votes of the classes are $\langle 0, 0.4, 1.6\rangle$. Since the class $C_3$ has received the highest amount of votes, 1.6, the class of the query instance is predicted to $C_3$. The confidence of this prediction is $1.6/2 = 80\%$.

## 3. FEATURE SELECTION USING A GENETIC ALGORITHM

Practical classification problems require the selection of a subset of features from a much larger set to represent the knowledge to be used in the classification. This is due to the fact that the performance of the classifier and the cost of classification are sensitive to the choice of the features used in the construction of the classifier. With the reduced set of features, the time needed for learning the classification knowledge and the time required for classification is reduced.

Further, by the extraction of relevant features and therefore the elimination of the irrelevant ones, the accuracy of the classifier can be increased [4, 16].

Exhaustive evaluation of possible feature subsets is usually infeasible in practice since it requires large amount of computational effort. Genetic Algorithms (GAs) offer an attractive approach to find near-optimal solutions to such optimization problems [6, 15, 20]. GAs are randomized search and optimization techniques guided by the principles of evolution and natural genetics, with a large amount of implicit parallelism [9]. In GAs, the parameters of the search space are encoded in the form of strings, called *chromosomes*. A collection of such strings is called a *population*. In the case of feature selection problem, each chromosome represents a subset of features selected. The size of a chromosome is equal to the number of features. Each element of the chromosome string is either 1 or 0, where 1 indicates that the corresponding feature is selected, and 0 otherwise. The goal of the search, in this case, is to find a chromosome that represents a set of features that lead to highest accuracy. In the case of several feature subsets with the same best accuracy, the one with the smallest cardinality is the desired one.

Initially a random population is created, representing different points in the search space. Each of the initial population are evaluated according to the fitness function. In the GA used in the experiments, the cube of the five-fold cross-validation accuracy is used as the fitness value of a chromosome. Then, until a maximum number of generations is reached, the following three operations are executed in order at each generation of the GA search: reproduction, crossover, and mutation. The GA used here employs the roulette-wheel selection in the reproduction step. As the crossover operation two-point crossover is used. After the generation of a new population, all the chromosomes created or mutated are evaluated again. The best chromosome is always copied to the next generation (*elitism*) by passing the reproduction step. The best chromosome is the one with the highest fitness value. Among the chromosomes that have the same fitness value, the one with the smallest number of features is chosen. The values for the parameters of the GA used in experimentations are given in Section 5.

## 4. ARRHYTHMIA DETECTION

The dataset used here consists of 533 ECG records recorded from 452 subjects (203 males with age $48 \pm 17$; 249 females with age $46 \pm 16$). Each record consists of a set of clinical parameters measured on rest ECG signals (Figure 4) automatically by a commercially available system[1], and some personal information about the subjects. There are 279 parameters (features) in a single record.

[1]Kardiosis$^{TM}$ system of TEPA A.Ş., Ankara, Turkey

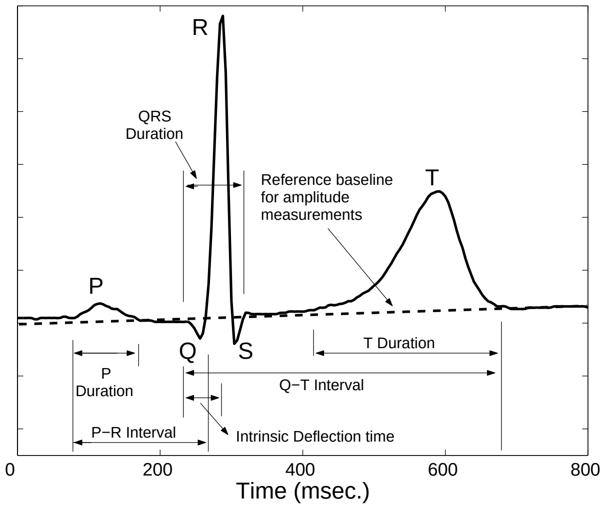


Figure 4: Time interval measurements done on a heart beat.

The patient population is divided into two groups based on the investigation of an expert cardiologist as *Normal* and *Abnormal*, represented by classes $C_1$ and $C_2$, respectively. The cardiologist was provided with the graphical plots of the ECG waveforms and the available personal information about the patient, i.e. age, height, weight and sex. There are 245 cases in the normal group and 288 cases in the abnormal group. The abnormal group consists of the following abnormalities: Ischemic Changes, Old Anterior Myocardial Infarction, Old Inferior Myocardial Infarction, Sinus Tachycardy, Sinus Bradycardy, Ventricular Premature Contraction(PVC), Supraventricular Premature Contraction, Left Bundle Branch Block, Right Bundle Branch Block, Left Ventricule Hypertrophy, Atrial Fibrillation and Flutter.

Out of 279 features 206 of them are continuous valued (linear) and 73 features are boolean valued (nominal). The first four features ($f_1 \cdots f_4$) are age, sex, height and weight, respectively. The feature $f_5$ is the average QRS duration in milliseconds, while $f_6$ is the average time interval between the onset of P and Q waves. The features $f_{10}$ to $f_{14}$ are the vector angles in degrees on the front plane of $QRS$ ($f_{10}$), $T$ ($f_{11}$), $P$ ($f_{12}$), $QRST$ ($f_{13}$), and $J$ ($f_{14}$), respectively. The feature $f_{15}$ represents the heart rate in terms of beats per minute. The next 11 features ($f_{16}$-$f_{27}$) are measured in lead DI: $f_{16}$: Average duration of Q wave; $f_{17}$: Average duration of R wave; $f_{18}$: Average duration of S wave; $f_{19}$: Average duration of R' wave (the small amplitude positive deflection just after the R wave, which is observed in some recordings); $f_{20}$: Average duration of S' wave (the small amplitude negative deflection just after the S wave, which is observed in some recordings); $f_{21}$: Intrinsic deflection time (i.e. the ventricular activation time). Features $f_{22}$ through $f_27$ are nominal valued. The last 12 features are also measured on lead DII ($f_{28}$-$f_{39}$), DIII ($f_{40}$-$f_{51}$), AVR ($f_{52}$-$f_{63}$), AVL ($f_{64}$-$f_{75}$), AVF ($f_{76}$-$f_{87}$), V1 ($f_{88}$-$f_{99}$), V2 ($f_{100}$-$f_{111}$),

Table 1: Classifications when all features are used.

| | Predicted as | |
|---|---|---|
| Actual | Abnormal | Normal |
| Abnormal | 195 | 93 |
| Normal | 47 | 198 |

V3 ($f_{112}$-$f_{123}$), V4 ($f_{124}$-$f_{135}$), V5 ($f_{136}$-$f_{147}$), V6 ($f_{148}$-$f_{159}$). In addition to the time domain measurements defined above, the following amplitude measurements are also done: $f_{160}$: J point depression on DI in mV.; $f_{161}$: Q wave amplitude on DI in mV.; $f_{162}$: R wave amplitude on DI in mV.; $f_{163}$: S wave amplitude on DI in mV.; $f_{164}$: Amplitude of R' on DI in mV.; $f_{165}$: Amplitude of S' on DI in mV.; $f_{166}$: P wave amplitude on DI in mV.; $f_{167}$: T wave amplitude on DI in mV.; $f_{168}$: QRS Area (Sum of areas of all segments divided by 10. The area of a segment is defined as the product of its time duration and amplitude divided by 2); $f_{169}$: QRST Area (= $QRS\_Area + 0.5 \times T\_duration \times 0.1 \times T\_amplitude$). These last 10 features are also measured on other leads in the same order ($f_{170}$-$f_{279}$).

In the dataset used in the experiments 0.33% of the feature values are missing. However, as explained in Section 2, the CFI algorithm is capable of handling such a missing data set.

## 5. EXPERIMENTS ON THE ARRHYTHMIA DATASET

In order to determine the set of relevant features we used a GA as explained in Section 3. In this experiment, the GA had 500 chromosomes, and each chromosome had 279 binary valued (0 and 1) genes, one for each feature. The value 1 represented the fact that the corresponding feature is selected, and vice versa. The GA used two-point crossover, with the probability of crossover $p_c = 0.8$. The probability of mutation was $p_m = 5.10^{-5}$. The GA was run for 1000 generations.

As the fitness function, the cube of the 5-fold cross-validation accuracy of the CFI algorithm using the set of features selected by the corresponding chromosome is used. The reason for using the cube function is to expand the gap between the fitness values for chromosomes with above the default accuracy.

The cost of misclassification is not symmetric between the two classes, normal and abnormal, in the case of arrhythmia detection. That is, the cost of misclassifying an abnormal patient as normal has a higher cost then misclassifying a normal patient as abnormal. Considering this fact we have defined the accuracy of CFI as

$$accuracy = \frac{a_a + n_n - a_n}{a_a + a_n + n_a + n_n}$$

here, $a_a$ denotes the number of abnormal cases predicted as abnormal, while $a_n$ denotes the number of abnormal cases predicted as normal. Similarly, $n_a$ and $n_n$ are the number of normal cases classified as abnormal and normal, respectively.

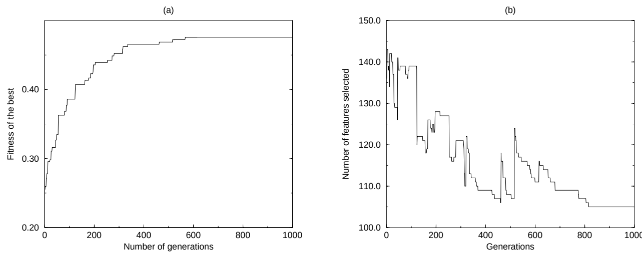


Figure 5: (a) The fitness and (b) the number of features selected for the best chromosomes.

Table 2: Classifications when only selected features are used.

| | Predicted as | |
|---|---|---|
| Actual | Abnormal | Normal |
| Abnormal | 244 | 44 |
| Normal | 29 | 216 |

In order to compute the 5-fold cross-validation accuracy, the whole dataset is partitioned into five equal size subsets. The four of the subsets is used as the training set, and the fifth one is used as the test set. This process is repeated five times, once for each subset being the test set. The final accuracy is the average of the accuracies obtained in these five runs. This technique ensures that each case is used exactly once in the test set.

We first experimented with the CFI on the arrhythmia dataset using all features (no feature selection). The CFI algorithm achieved 56.29% accuracy. The training time for each fold was 208 msec, while the testing time was 37 msec. The classification table for all features is given in Table 1.

Then, we ran the GA specified above to find a good set of relevant features, so that the accuracy of CFI can be increased. The best fitness values and the number of features selected for the best chromosomes through out the execution of the GA are shown in Figure 5. At the end of the 1000th generation of the GA, the best chromosome contains only 105 features out of 279. The accuracy of the CFI algorithm with this set of features is 78.05%. Using only these 105 relevant features, the training time for each fold was 80 msec, while the testing time was 16 msec. The classification table for selected features is given in Table 2. Using this set of features 86.3% of all cases are classified correctly.

## 6. CONCLUSIONS

In this paper, a new classification algorithm called CFI is developed and applied to the detection of abnormal ECG recordings. Since CFI treats each feature, the missing feature values that may appear both in the training and test instances are simply ignored. In other classification algorithms, such as decision tree inductive learning algorithms, the missing values require extra care [18]. This problem has been overcome by simply omitting the feature with the missing value in the voting process of CFI. Also note that the CFI algorithm is applicable to concepts where each feature, independent of other features, can be used in the classification of the concept. One might think that this requirement may limit the applicability of the CFI, since in some domains the features might be dependent on each other. Holte has pointed out that the most datasets in the UCI repository are such that, for classification, their attributes can be considered independently of each other [12]. Also Kononenko claimed that in the data used by human experts there are no strong dependencies between features because features are properly defined [13]. Another advantage of the CFI classifier is that instead of a categorical classification, a more general probabilistic classification where the classifier returns a probability distribution over all classes is possible to implement with CFI.

The original data set of ECG recordings that we used contained 279 features. In order to select and use only the relevant features, we developed a genetic algorithm. We found that only 105 features are sufficient for the detection of abnormal cases. Using only the revelant features increased the accuracy and decreased both the training and the prediction times of the CFI algorithm.

**Acknowledgments**
This project is supported, in part, by TUBITAK (Scientific and Technical Research Council of Turkey) under Grant EEEAG-153. The authors thank Prof. Hayrettin Köymen and Dr. Ayhan Çekin for providing the necessary equipment and data for the project.

# References

[1] D. W. Aha, D. Kibler and M. K. Albert, "Instance-based learning algorithms", Machine Learning, Vol. 6, 1991, pp. 37-66.

[2] A. Akkuş, and H. A. Güvenir, "K nearest neighbor classification on feature projections", Proc. of ICML'96, 1996, pp. 12-19.

[3] A. Akkuş, "Batch learning of disjoint feature intervals," *MSc. Thesis*, Bilkent University, Dept. of Computer Engr. & Info. Sci., 1996.

[4] H. Almuallim and T. G. Dietterich, "Learning boolean concepts in the presence of many irrelevant features", Artificial Intelligence, Vol. 69, 1994, pp. 279-305.

[5] S. Cost and S. Salzberg, A weighted nearest neighbor algorithm for learning with symbolic features", Machine Learning, Vol. 10, 1993, pp. 57-78.

[6] G. Demiröz and H. A. Güvenir, Genetic algorithms to learn feature weights for the nearest neighbor algorithm", Proc. of BENELEARN-96, 1996, pp. 117-126.

[7] G. Demiröz and H. A. Güvenir, "Classification by voting feature intervals", Proc. of Ninth ECML, (Springer-Verlag, LNAI 1224), 1997, pp. 85-92.

[8] J. Forsström, P. Eklund, H. Virtanen, J. Waxlax and J. Lähdevirta, "DIAGAID: a connectionist approach to determine the diagnostic value of clinical data", Artificial Intelligence in Medicine Vol. 3, 1991, pp. 193-201.

[9] D. E. Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning, Addison Wesley, Reading, Massachusetts, 1989.

[10] H. A. Güvenir, G. Demiröz and N. İlter, "Learning differential diagnosis of erythemato-squamous diseases using voting feature intervals", Artificial Intelligence in Medicine, Vol. 13, 1998, pp. 147-165.

[11] H. A. Güvenir and İ. Şirin, Classification by feature partitioning", Machine Learning, Vol. 23, 1996, pp. 47-67.

[12] R. C. Holte, "Very simple classification rules perform well on most commonly used datasets", Machine Learning, Vol. 11, 1993, pp. 63-91.

[13] I. Kononenko, "Inductive and Bayesian learning in medical diagnosis", Applied Artificial Intelligence, Vol. 7, 1993, pp. 317-337.

[14] I. Kononenko and I. Bratko, "Information-based evaluation criterion for classifier's performance", Machine Learning, Vol. 6, 1991, pp. 67-80.

[15] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers", Pattern Recognition, Vol. 33, 2000, pp. 25-41.

[16] H. Liu and R. Setino, "A probabilistic approach to feature selection - A filter solution", Proc. of the 13th ICML, 1996, pp. 319-327.

[17] B. Lopez and E. Plaza, "Case-based learning of plans and goal states in medical diagnosis", Artificial Intelligence in Medicine, Vol. 6, 1997, pp. 29-60.

[18] J. R. Quinlan, "Unknown attribute values in induction", Proc. of the 6th International Workshop on Machine Learning, 1989, pp. 164-168.

[19] A. K. Spackman, "Learning categorical decision criteria in biomedical domains", Proc. of the Fifth ICML, 1988, pp. 36-46.

[20] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm", Feature Extraction, Construction and Selection - A Data Mining Perspective, Motoda and Liu (Eds). Kluwer Academic Publishers, 1988.