Analysis and Diagnosis of Erythemato-Squamous Diseases using CHAID Decision Trees

Alaa M. Elsayad
College of Engineering at Wadi Addawaser,
Prince Sattam Bin Abdulaziz University, KSA.
Computers and Systems Department, Electronics Research Institute, Giza, 12622, Egypt
Email: sayad@eri.sci.eg

Mujahed Al-Dhaifallah
College of Engineering at Wadi Addawaser,
Prince Sattam Bin Abdulaziz University, KSA.
Systems Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, 31261, KSA.
Email: mujahed@kfupm.edu.sa

Ahmed M. Nassef
College of Engineering at Wadi Addawaser,
Prince Sattam Bin Abdulaziz University, KSA.
Department of Computers and Automatic Control Engineering,
Faculty of Engineering, Tanta University, Egypt.
Email: ahmed_nassef2004@yahoo.co.uk

***Abstract*— Erythemato-squamous diseases (ESDs) are common skin diseases. They consist of six different categories: psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis and pityriasis rubra pilaris. They all share the clinical features of erythema and scaling with very little differences. Their automatic detection is a challenging problem as they have overlapping signs and symptoms. This study evaluates the performance of CHAID decision trees (DTs) for the analysis and diagnosis of ESDs. DTs are nonparametric methods with no priori assumptions about the space distribution with the ability to generate understandable classification rules. This property makes them very efficient tools for physicians and medical specialists to understand the data and inspect the knowledge behind. The Chi-Squared Automatic Interaction Detection (CHAID) decision tree model is a very fast model with the ability to build wider decision trees and to handle all kinds of input variables (features). The CHAID model has many successful achievements especially when used as an interpreter rather than a classifier. Due to the small number of samples, this study uses Chi-square test with the Likelihood Ratio (LR) to get robust results. Ensembles of bagged and boosted CHAIDs were introduced to improve the stability and the accuracy of the model, but on the expense of interpretability. This paper presents the experimental results of the application of CHAID decision trees and their bagged and boosted ensembles for the deferential diagnosis of ESD using both clinical and histopathological features. The prediction accuracies of these models are benchmarked against the Artificial Neural Network (ANN) in terms of statistical accuracy, specificity, sensitivity, precision, true positive rate, true negative rate and F-score. Experimental results showed that bagged ensemble outperforms other modeling algorithms.**

Keywords—: erythemato-squamous diseases, automatic differential diagnosis, decision tree, bagging, boosting, CHAID, Artificial Neural Network, multi-class classification

I. INTRODUCTION

Erythemato-squamous diseases (ESDs) are popular classes of dermatological infections. There are six different groups of ESDs. In the outpatient dermatology departments, these groups of the disease are known as psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis and pityriasis rubra pilaris [1, 2]. These diseases usually produce redness of the skin (erythema) caused by the loss of skin cells (squamous). Typically, they have either genetic or environmental causes - and tend to occur at specific periods of life such as late childhood/early adolescence. Their differential data analysis is a difficult problem in dermatology because one disease may indicate features of another disease at the initial stage and then it has its own characteristics at the following stages. Moreover, they share many clinical and histopathological features, which make the problem more difficult. To diagnose the disease, usually patients are checked clinically and histopathologically [3]. The clinical tests are performed with non-invasive examinations of the symptoms - such as location, size, and presence of pustules, color, and related features. The Histopathological tests require extracting skin samples (biopsies) for detecting the potential viral sources. From the literature, it has been found that the differential diagnosis of ESDs is widely discussed as a data-mining problem [4, 5].

In general, data mining refers to the algorithm of automated extraction of hidden, previously unknown and hypothetically valuable information from large dataset. Data mining model works to find and interpret the valuable information based on multidisciplinary fields such as statistics, artificial intelligence, machine learning, database management, etc. [6]. In data mining field, there are lots of models and algorithms. Some of these models are able to give understandable decisions and the others are not. There is a trade-off between model accuracy and model transparency. In some applications, the accuracy of classification or prediction is the only thing that matters. In

978-1-5386-5305-0/18/$31.00 ©2018 IEEE

other situations, the ability to explain the reasons for taking a specific decision is the very crucial thing.

While the predictive accuracy achieved by ANN is often higher than that obtained by other models, still they have no ability to explain how they reach this particular conclusion [6]. This is definitely due to the high complexity of their structures [7]. On the other hand, decision trees (DTs) are powerful and popular for both data classification and prediction. Additionally, they have the transparent nature. DTs are considered as supervised algorithms for classification and regression. They partition the dataset recursively to form homogeneous groups or classes. The attractiveness of tree-based methods is largely due to the fact that DTs represent rules. Rules can readily be expressed in simple words or graphical representation so that humans can understand them easily [8].

This paper investigates the applicability of CHAID decision tree models for the differential diagnosis of ESDs and compares their predictions to those generated by ANN. CHAID stands for Chi-Square Automatic Interaction Detector, introduced by Kass in 1980 [9]. The novel inspiration for CHAID model was for detecting statistical relationships between input and output variables. It does this by building a decision tree, so the method has become as a classification tool as well. The CHAID model relies heavily on the Chi-Square statistical test in all modeling steps. The algorithm generates wider trees and hence it provides a good compact explanatory decision tree. However, CHAID, like other decision tree models, has a problem of stability [10]. Bagging and boosting are two different aggregation algorithms to build tree-based ensembles. Bagging method enhances model stability while boosting increases model accuracy. However, these tree-based ensembles provide good prediction solution on the expense of interpretability.

The remainder of the paper is organized as follows: Section II emphasizes on the clinical and histopathological features of ESDs and introduces the recent related literatures. Section III reviews the classification models CHAID, bagging, boosting and Multi-layer neural network. Section IV demonstrates the implementation of proposed stream with SPSS modeler software from IBM©. Experimental results and discussion are also presented to show the effectiveness of the proposed ensemble. Finally, conclusions are introduced in Section V.

II. DERMATOLOGY DATASET AND RELATED WORK

Dermatology dataset was designed to determine the types of ESDs. It can be obtained from the University of California at Irvine (UCI) machine learning repository [11]. The dataset contains 366 samples with 34 input and one output (target) variables. The number of samples is reduced to 358 after the removal of 8 samples because they have missing values. The remaining samples (358) are belonging to six different classes (diseases). In clinical diagnosis, patients are firstly examined clinically with 12 different tests. However, they are not sufficient and dermatologists request a biopsy to derive 22 histopathological features for every sample. Each dermatology subject has a data vector of 34 features. All clinical and histopathological features, except that age and family history, are evaluated with a degree in the range of 0–3. If the feature does not exist it takes 0 (NON), however it takes 3 (maximum) when the largest possible value occurs. For relative intermediate values 1 or 2, it refers to Low or Medium, respectively. For the family history feature, it takes value 1 if any of these diseases are observed in a family member; otherwise, it takes the value of 0. Each feature represents either ordinal (discrete), linear (continuous), or nominal (binary) variable. Table I shows the 34 features and the six classes of the diseases used in this study.

TABLE I CLASS DISTRIBUTION

Class number	Erythemato-squamous diseases	Number of samples
1	Psoriasis	111
2	Seboreic dermatitis	60
3	Lichen planus	71
4	Pityriasis rosea	48
5	Chronic dermatitis	48
6	Pityriasis rubra pilaris	20

The dataset is a donation from Güvenir et. al. which is firstly published in 1998 [2]. The donors built an expert system that incorporated two predicting models; Nearest Neighbor classifier and Naive Bayes classifiers with voting feature intervals-5 [1]. Recently, several scientific articles have been released showing the results of different data mining tools for the diagnosis of ESDs. A very recent paper applied hybrid methods that use Granular Computing (GrC) and support vector machines (SVM) [12]. GrC algorithm is derived from rough sets theory with the ability to describe the real world problem in different levels of granularities; thereby it can abstract the problem at different levels of abstractions. The authors reviewed and evaluated most of the past artificial intelligence systems used for the diagnosis of ESDs and they tabulated the classification results of all these algorithms. Their results achieved averages of sensitivity and specificity as 98.43%, and 99.71%, respectively. The comparisons made by these authors showed that the decision tree algorithms and other probability estimation models have been used very few times comparing to the other black box models, e.g. ANN and SVM. In addition, our extensive reviews in recent literatures match with this conclusion. Recently, classification and regression trees (CART) have been applied for the same purpose using the SPSS© Clementine software from IBM® and they also achieved an accuracy of 94.84%. Liu et. al. applied the C4.5 decision tree and obtained an accuracy of 95.08% [13].

III. CLASSIFICATION ALGORITHMS

A. Decision tree models

Decision tree models are machine-learning algorithms used for classification and prediction applications. They have a tree-like graphical representation that consists of collections of branches, decision nodes and terminal nodes (leaves). Branches express the values of the relevant features and the leaves represent the values of the output classes. A tree path represents a rule to produce a classification or generate a

prediction. The popular decision tree models are CHAID, CART, C5.0 and MARS. CHAID, Chi-Square Automatic Interaction Detector, is proposed by Kass in 1980 [9]. CART, Classification and Regression Trees, is proposed by Breiman et. al. in 1984 [14]. MARS, Multi-Adaptive Regression Splines [15]. Finally, C4.5 and its most recent version C5.0 is proposed by Quinlan in 1999 [16, 17]. The main differences between these models are the splitting criterion, the allowable data types of the input variables, the ability to handle missing data, the pruning algorithm and finally the ability to build regression and/or classification algorithm. This study used CHAID for the diagnosis of the ESDs. The advantages and disadvantages of CHAID model is demonstrated and investigated in [18-20]. CHAID is proved that it is the most appropriate technique for selecting the more meaningful or important segmentation variable as an intermediate step for proper data segmentation, especially for demographic or behavioral data [21].

1) CHAID Decision tree model

CHAID algorithm generates decision trees using Chi-Square statistics to define the optimal splits [9]. Unlike the CART model, CHAID has the ability to produce non-binary trees, i.e., some splits may have more than two branches that is particularly well suited for the analysis of larger datasets. The algorithm transforms continuous features into ordinal ones using binning algorithm, as it works only with nominal or ordinal (categorical) inputs. The learning process works using a heuristic statistical method to check the relation between the set of categorical features and the target output. It produces a tree graph that discloses the categories of features that most significantly predict the value of target output. CHAID modeling algorithm is as follows [22, 23]:

1. *Binning step*: It is the first step of the CHAID learning algorithm to transform all continuous features into categorical ones using binning process. For a continuous feature x with a given set of break values a_1, a_2, a_3,…, a_{K-1} arranged in an ascending order, the criterion that categorizes the features $C(x)$ will be as follows (1):

$$C(x)=\begin{cases}1 & x\le a_1\\ k+1 & a_k<x\le a_{k+1},\ k=1,...,k-2\\ k & a_{k-1}<x\end{cases} \quad (1)$$

2. *Merging step*: The second step is to reduce the number of categories of every feature. For every feature, the algorithm merges any pair of categories that shows no differences with respect to the output variable by applying a Chi-Square test (with Pearson Chi-Square or Likelihood ratio) [24]. Then, the algorithm computes the *p*-value (significance value). Any two categories are merged into a single one if the *p*-value is greater than α_{merg}; where, α_{merg} is a prespecified value in the range]0, 1]. The value must be greater than 0 and less than or equal to 1. However, if the value of α_{merg} is set to 1, there will be no allowable merging step at all. This process is repeated successively to combine every pair of categories that shows the least important difference. The categories of nominal features may be merged with no restrictions; but for an ordinal feature (categories that have intrinsic ranking) only adjacent categories can be merged. The CHAID algorithm computes the adjusted *p*-value for the merged categories by applying Bonferroni adjustments [24]. The Bonferroni multiplier B is the number of potential ways that I categories can be merged into r categories. For $r = I$, $B = 1$. For $2 \le r < I$, the following equation is used:

$$B=\begin{cases}\binom{I-1}{r-1} & \textit{Ordinal Predictor}\\ \sum_{v=0}^{r-1}(-1)^v\frac{(r-v)^I}{v!(r-v)!} & \textit{Nominal Predictor}\\ \binom{I-2}{r-2}+r\binom{I-2}{r-2} & \textit{Ordinal with missing category}\end{cases} \quad (2)$$

3. *Splitting step*: the algorithm searches for the split point (best one) with the smallest adjusted *p*-value to find the best split. The searching process uses the adjusted *p*-value to examine the association of every feature with the output variable. The feature with the lowest adjusted *p*-value wins the strongest association value. If this value is less than or equal a pre-specified split threshold α_{split}, this feature is selected as the split feature for the current node. After splitting process, the resulting child nodes are evaluated whether if they are qualified for further splitting or not.

4. *Stopping step*: The merge/split processing continues repeatedly until one or more stopping rules are met for every unsplitted node and hence no further splits can be applied.

This study uses the Likelihood ratio to compute the Chi-Square test [24]. The algorithm forms a contingency table which contains the classes of the output variable y as columns and the categories of the input features x as rows. The expected cell frequencies under the null hypothesis independence (i.e. no relation between x and y) are evaluated. The observed cell frequencies and the expected cell ones are used to calculate the Chi-Squared statistic and the *p*-value.

$$G^2=\sum_{j=1}^{J}\sum_{i=1}^{I}n_{ij}\ln\left(\frac{n_{ij}}{\hat{m}_{ij}}\right) \quad (3)$$

where $n_{ij}=\sum_n f_n I(x_n=i\wedge y_n=j)$ is the observed cell frequency and is the expected cell frequency for cell $(x_n=i, y_n=j)$, and the p-value is computed as follows:

$$p=\Pr\left(\chi_d^2>G^2\right) \quad (4)$$

The advantages of this model are as follows:

- It is based on the Chi-Square statistics;
- It is a non-parametric statistical model of free distribution;
- No type of distribution is assumed a priori;
- It can handle all types of input variables: binary, nominal, ordinal and continuous;
- It is very fast with the ability to build "wider" decision trees with the ability to summarize knowledge; making it very popular in several applications;
- The model uses a pre-pruning strategy- a node is only split if a significance criterion is fulfilled.

Despite the above advantages, this algorithm requires a large number of training data to obtain trustable results. Also, it has a problem of bias toward variables with more levels for

splits, which can skew the interpretation of the relative importance of the feature in explaining the responses of the target variable [14]. Additionally, CHAID, like other decision tree models, suffers from the problem of instability [8].

2) Bagging and boosting ensembles

An ensemble model works by training a set of individual classifiers (base models) and then combining their predictions using a certain aggregation rule. If base models are both accurate and diverse, the aggregation results will lead to higher accuracy than base models [25, 26]. In particular, the success of the ensemble models depends mainly on the diversity property. This diversity may happen when the ideas of model construction are different, *e.g.*, by combining classification trees, logistic regression and neural network), or by using the single algorithm and resampling the training set (like in bagging, boosting or random subspace procedure). Bagging and boosting are the most widely used ensemble algorithms.

Bootstrap aggregating ('bagging') is first appeared by Breiman in [14], where bootstrap sample is derived B times from the training data with the possible replacements, where N is the size of training set. An individual CHAID model is trained on each of the B bootstrap subset. The predictions of the resulting models are combined together with unweighted majority vote. All the training processes are working in parallel and may be executed on different computers. On the contrary, boosting ensemble is constructed in series. It begins by training the first base classifier (CHAID in our case) and then samples that are hardly modeled are given more weight in consecutive iteration using a multi-stage adjustment strategy [27]. The final decision is augmented using weighted majority vote based on base classifiers' performance. Boosting approach is originated from Freund & Schapire in [28] using resampling and combining algorithms to increase the weights of misclassified samples. AdaBoost (Adaptive Boosting) is the most common boosting algorithm [29]. This algorithm permits the designer to continue adding classifiers to the ensemble up to achieving certain error.

Bagging algorithm

Initialize:

- Consider we have the training set (x_i, y_i), $i = 1, \ldots, N$, of input vectors x_i and class labels y_i.
- Let B is the number of bootstraps versions of the training data

Training:

- For $b = 1, \ldots, B$, do the following: (b-the number of loop)
  - Sample with replacement N samples from the training set. Some samples will be replicated, others will be omitted.
  - Design a classifier, $K_b(x)$.

Testing:

- Produce the final decision for a test pattern x by recording the class predicted by $K_b(x)$, $b = 1, \ldots, B$, and find the majority vote.

AdaBoost Algorithm

Initialize:

- Consider we have the training set (x_i, y_i), $i = 1, \ldots, N$, of input vectors x_i and class labels y_i.
- Let M is the number of classifiers
- Initialize: each input vectors with an equal weight w_i

$$w_i = 1/N,\ i \in \{1, \ldots, N\}$$

Training

- For $m = 1, \ldots, M$, do the following: (m-the number of loop)
  - construct a classifier $K_m(x)$ from the training data with the weights w_i, $i = 1, \ldots, n$;
  - calculate e_m error as the sum of the weights w_i for the misclassified samples;
    - if $e_m > 0.5$ or $e_m = 0$ then terminate the procedure,
    - otherwise set $W_j = w(1 - e_m)/e_m$ for the misclassified samples and;
    - recalculate the weight distribute for the next classifier so that they sum to unity.

Testing:

- Each Sample is classified via the so-called 'weighted majority voting'.

B. *Artificial neural network*

Artificial Neural Networks (ANNs) are the most popular tools in classification and regression applications. ANN has shown effective results in approximating complex and nonlinear functions with no priori assumption to the model characteristics or data distribution [30, 31]. ANN conventionally consists of computational units (called neurons) in three or more layers. Neuron accumulates its input data x_i after multiplying them by the appropriate strengths of the respective connection weights w_{ij}. Then, the neuron fires its output y_j using activation function (AF) as shown in Fig. 1.

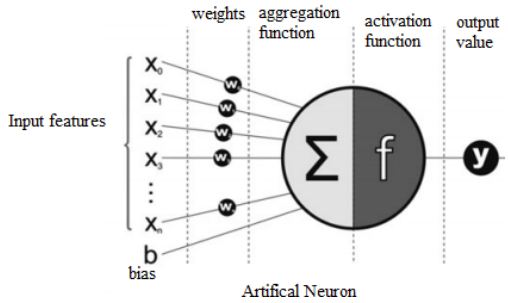


Fig. 1. Artificial Neuron

The weight values represent the priority and the significance of each input feature comparing to the others. AF can be in the form of a simple threshold function, sigmoidal, hyperbolic tangent, or radial basis function (RBF). The mathematical description of an artificial neuron can be written as equation (5).

$$y_i = f(\sum w_{ij} x_i), \tag{5}$$

where f is the activation function.

These artificial neurons are connected together to form a processing structure. A simple neural network has a feed-forward structure: the stream flows from inputs, forwards through any hidden units, finally reaching the output elements. Multilayer perceptron neural network with back-propagation is the most popular artificial neural network architecture [31]. Fig. 2 shows the structure of three layers ANN model which is organized into layers of input, hidden and output layers. Back-propagation (BP) is a popular learning algorithm for ANN. BP works by feeding training samples one-by-one to the network and then finds the squared error between the real output and network output (estimated) as follows:

$$E = \frac{1}{2}\sum_j (y_{dj} - y_j)^2 \tag{6}$$

where y_{dj} is the desired value of output neuron j and y_j is the estimated output of that neuron.

BP corrects iteratively the network weights and biases to reduce the summation of all squared errors; *i.e.*, the neural network is able to learn, and can thus reduce the future errors. The performance of ANN depends on the network configuration, weights, biases as well as the activation functions.

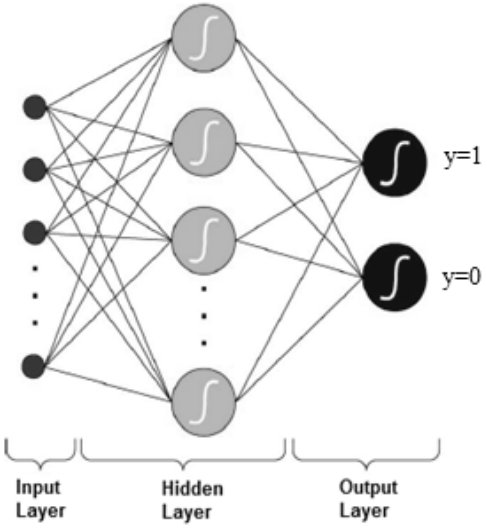


Fig. 2. Graphical representation of multilayer perceptron neural network for two class (binary) problem

Despite their popularity, the optimal structure of the ANN is still a challenging task and is not easy to solve. On one side, the structure has to be relatively small to increase the generalization. On the other side, the network has to classify the training samples efficiently. Therefore, some training algorithms start with a large network and then apply a pruning method to remove redundant connections, which finally results in a smaller network without sacrificing its performance [32].

IV. Expremental Results

Fig. 3 shows the component nodes of the proposed stream. The stream is implemented using IBM® SPSS Modeler data mining workbench [24]. The stream represents the sequential processes of developing and evaluating the mentioned predictive models for the analysis and diagnosis of ESDs. It starts by reading input data from SPSS file format, defining variable types with the *type* node, auditing the data to explore the basic statistics of each feature and checking the quality of the input data. Then processes continue by partitioning the input data into training and testing subsets, balancing the uneven class distribution, developing the predictive models on the training subset, validating the developed models, and finally scoring the testing data set.

A. *Data Exploration and Preprocessing*

The *Input* node is attached directly to SPSS file to provide the training and testing data to the subsequent nodes. This node explored the data for incorrect, inconsistent or missing data. There are eight samples contain missing values for the age feature and they are eliminated from further processing. The *Automated Data Preparation* node to normalize the continues features' values to have zero mean and 1 standard deviation using the z-score transformation $z = (x - \mu) / \sigma$, where x is the feature value, μ is mean value and σ is the standard deviation. Here in this study, the final mean is set to 2 to match with the dynamic range of other variables.

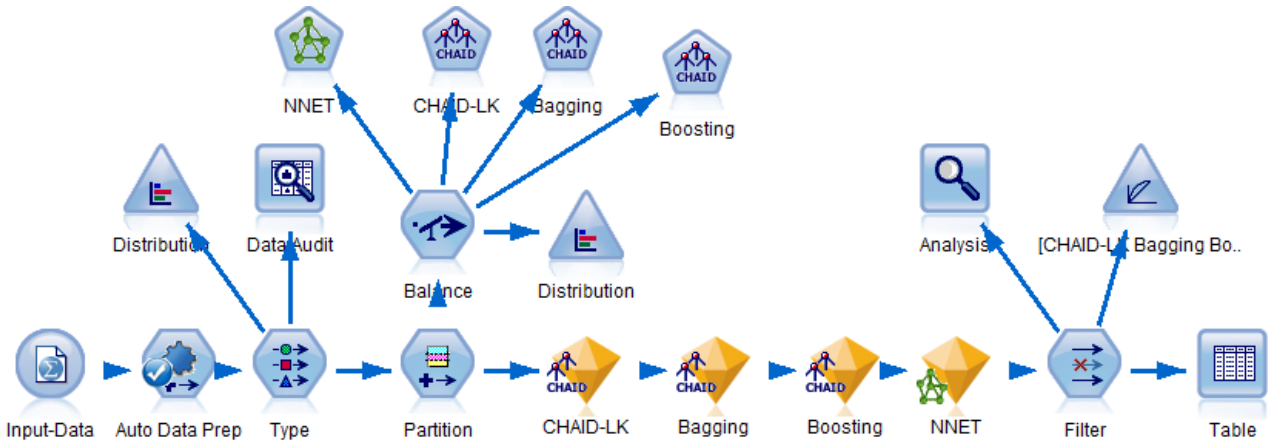
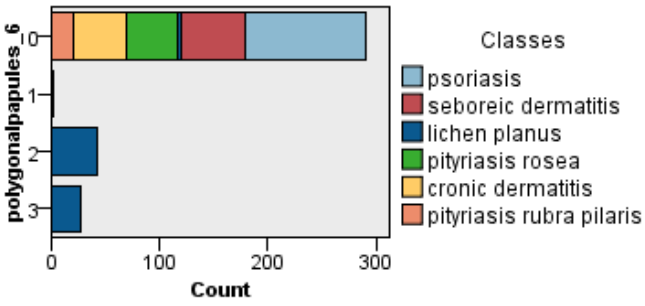
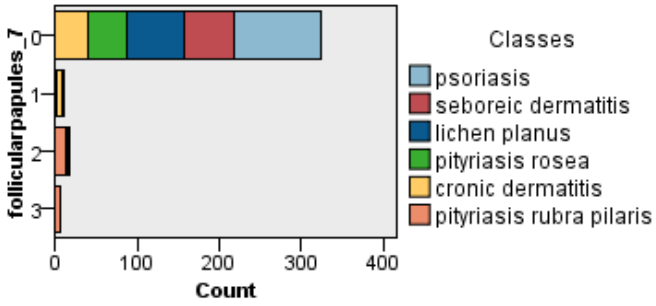


Fig. 3. Stream of analysis and prediction models for differential diagnosis of Erythemato-Squamous Diseases

The *Type* node sets the field metadata, whether it is for input or output class and all required properties that are important to construct the predicting models. The ESDs dataset contains 34 input features and only one target class variable. The *Data Audit* node provides a comprehensive statistical analysis for all features and provides summary statistics, histograms, and distribution graphs that is useful in gaining a preliminary understanding of the data. Fig. 4 shows two examples of feature distribution of "Polygonal papules' and "Follicular papules" among different classes. Fig. 4 (a) shows that feature "Polygonal papules" takes the values of 1,2, and 3 with only class "Lichen planus". While Fig. 4(b) shows that feature "Pityriasis rubra pilaris" takes the value 4 when the case is "pityriasis rubra pilaris". Such graphical exploration tools increase the medical information, allows deep data understanding and diagnosis knowledge transfer with dermatologists.



(a)

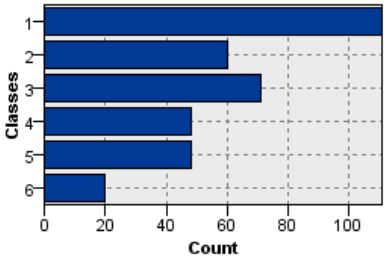


(b)

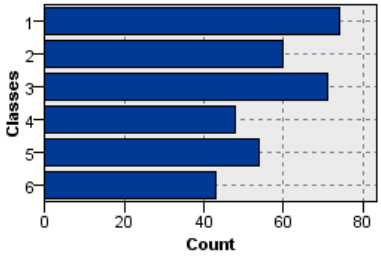
Fig. 4. Two examples of feature values distributions (histogram) and how these values are distributed to different disease groups

B. *Partitioning and balancing*

The *Partition* node divides the dataset into training and testing subsets by the ratio 70:30, respectively. The *Balance* node corrects the unevenness in the training subset to some extent. Table I shows the unequal distribution of samples among different groups. Typically, classification models tend to learn the majority classes more perfectly than the minority ones [33]. The *Balance* node can be used to set the conditions required to perform under-sampling and over-sampling. Under-sampling reduces the number of samples from majority class and over-sampling duplicate the minority ones to make the training dataset balanced. On the other hand, samples that do not meet any conditions are allowed it to pass. Two *Distribution* nodes have been used to show the occurrence of different groups before and after the *Balance* node as shown in Fig. 5.



(a) Before the *Balance* node



(b) After the *Balance* node

Fig. 5. Class distribution before and after the *Balance* node

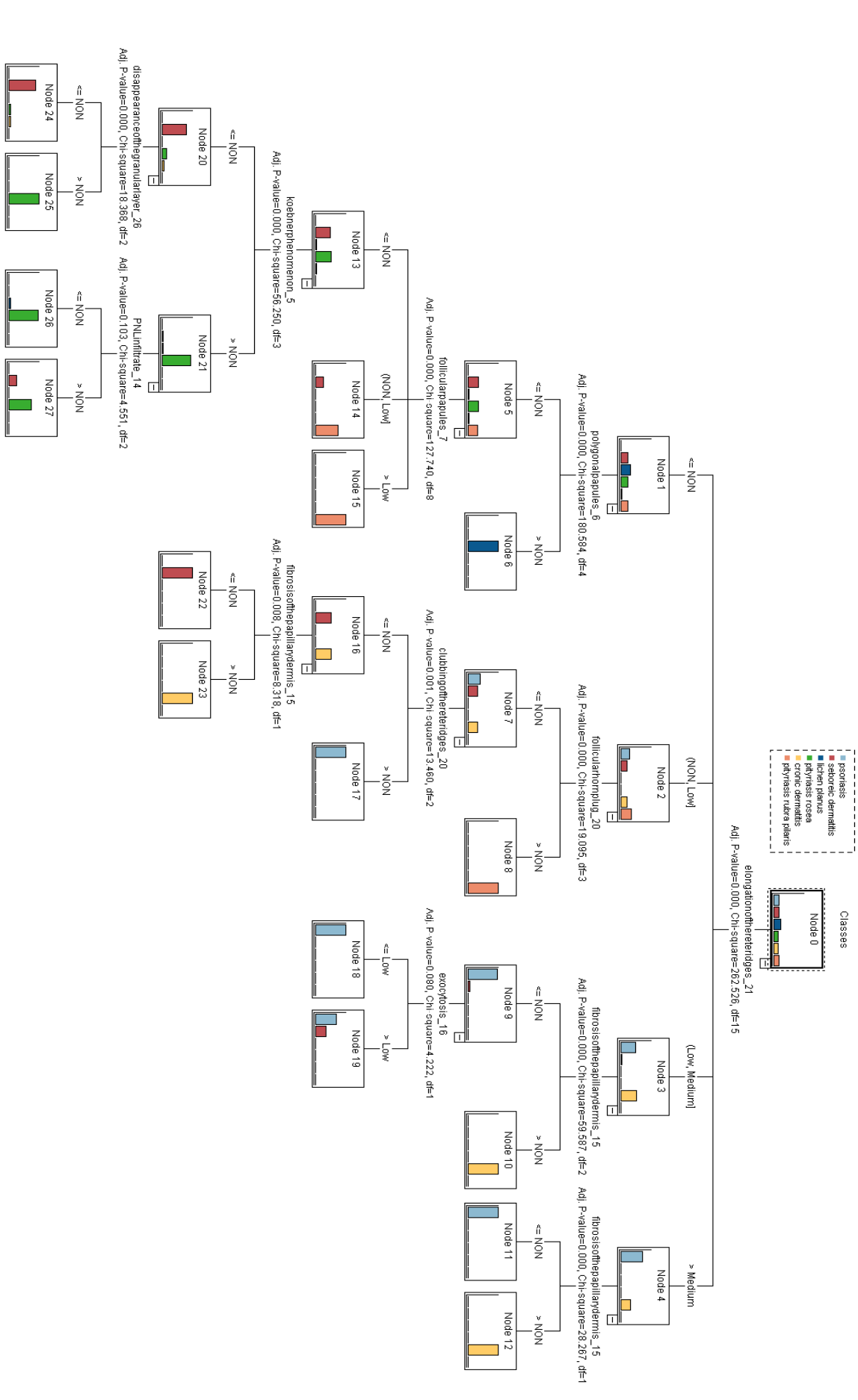


Fig. 6. Trained CHAID decision tree with Likelihood ratio.

C. Model Construction

The *CHAID* classifier node has been trained with the Likelihood Ratio to compute the Chi-Square for categorical targets as the number of training samples is relatively small [34]. The Chi-Square statistic is applied to merge categorical values that are statistically homogeneous with respect to the target variable. In this study, the output variable is of categorical type (it has six categories). The best significance level for splitting (α_{split}) and significance level for merging (α_{merg}) is found to be (0.15) and the maximum allowable depth is set to 5. After merging similar categories, the algorithm selects the best predictive feature to form the first branch in the decision tree, such that each child node is made of a group of homogeneous values of the selected attribute. The resulting accuracies for training and testing are 95.92% and 90.27%, respectively. The CHAID with likelihood ratio model is very fast; it takes below one second to build the model and easy to interpret; it can be expressed as a set of rules. Fig. 6 shows the trained decision tree with CHAID algorithm. It is obviously not a binary tree. Only ten features have been used only to build the model: 3 clinical and 7 histopathological while the age feature is not important.

The *Bagging* ensemble node is used to improve the stability of the single CHAID model and to avoid over-fitting. The number of component model is set to 50 in order to obtain predictions that are more reliable. Both α_{split} and α_{merg} are kept as those of the single CHAID model, however the tree depth is increased to 10. That is to increase the diversity among constituent models. All trees are pushed to fully grow without pruning and at each node in the tree, the algorithm searches over all features to find the feature that best splits the data at that node. The resulting accuracies for training and testing are improved significantly into 98.78% and 96.46%, respectively.

The *Boosting* ensemble node is used to improve the training accuracy, however it may achieve bad performance for testing subset. This problem is called over-fitting. The tree is kept stump as possible to trade-off between training accuracy and the generalization. The number of component models (boosting phases) is set to 10. Other tree specifications are remain the same as in single CHAID model. The accuracies obtained are 100% and 92.92% for training and testing with tree depth equal to 5.

NNET classifier node is trained using four layers; one input, two hidden and one output. The number of neurons in first and second hidden layers are set to 20 and 10, respectively. The training algorithm is unleashed to achieve 100% training accuracy. However, this optimistic accuracy never happen, the training process may be interrupted at any point to save the network model with the best accuracy achieved so far. Part of the training subset (30%) has been reserved for validation to increase the generalization and reduce the over-fitting. It takes only few seconds to achieve 98.37% and 93.81% for training and testing accuracies, respectively.

D. Model Evaluation

The multiclass confusion matrix shown in Table II has been used to evaluate and compare the performance of individual classification models more accurately [35]. It is a standard representation of the performance of classification algorithms. The table contains the number of correctly and wrongly classified samples compared to the real outputs in the training and testing data. This matrix allows more detailed examination than simple amount of correctly classified sample (accuracy) which may give misrepresentative results if the dataset is unbalanced as the case under investigation. This matrix has *N* by *N* size, where *N* is the number of classes. The ideal case when the true positive places (main diameter) have values while other matrix places contains only zeros. It has the ability to summarize the performance of individual classifier algorithm. It cross-tabulates predicted and actual samples into four options: true positive (*TP*), true negative (*TN*), false positive (*FP*) and false negative (*FN*). In this multiclass situation, they are defined separately for each class using *the one-against-all approach*. For example, for the "Lichen Planus" class these indicators are as follow:

1. *TP* of *"Lichen Planus"* is all "Lichen Planus" samples that are classified as *"Lichen Planus"*
2. *TN* of *"Lichen Planus"* is all *non* "Lichen Planus" samples that are *not* classified as *"Lichen Planus"*
3. *FP* of *"Lichen Planus"* is all *non* "Lichen Planus" samples that are classified as *"Lichen Planus"*
4. *FN* of *"Lichen Planus"* is all "Lichen Planus" samples that are **not** classified as *"Lichen Planus"*

Seven performance measures have been computed for testing data using equations (7-13). They are accuracy, sensitivity, specificity, precision, false positive rate, false negative rate, and F1-score.

$$\textit{Accuracy} \quad \frac{TP + TN}{N} \tag{7}$$

$$\textit{Sensitivity (True Positive Rate TPR)} \quad \frac{TP}{TP + FN} \tag{8}$$

$$\textit{Specificity (True Negative Rate TNR)} \quad \frac{TN}{TN + FP} \tag{9}$$

$$\textit{Precision(positive predictive value PPV)} \quad \frac{TP}{TP + FP} \tag{10}$$

$$\textit{False Positive Rate FPR} \quad \frac{FP}{TN + FP} \tag{11}$$

$$\textit{False Negative Rate FNR} \quad \frac{FN}{FN + PP} \tag{12}$$

$$\textit{F1 Score (Harmonic Mean of sensitivity and specificity)} \quad \frac{2TP}{2TP + FP + FN} \tag{13}$$

TABLE II MULTICLASS CONFUSION MATRIX FOR DERMATOLOGY DATASET: EXAMPLE TO COMPUTE TP, TN, FP, AND FN FOR "LICHEN PLANUS" USING ONE-AGAINST-ALL APPROACH.

Actual	Predicted						
	Psoriasis (1)	***Seboreic dermatitis (2)***	***Lichen planus (3)***	***Pityriasis rosea (4)***	***Chronic dermatitis (5)***	***Pityriasis rubra pilaris (6)***	
Psoriasis (1)	**TN**	**TN**	**FP_1**	**TN**	**TN**	**TN**	
Seboreic dermatitis (2)	**TN**	**TN**	**FP_2**	**TN**	**TN**	**TN**	
Lichen planus (3)	**FN_1**	**FN_2**	**TP**	**FN_4**	**FN_5**	**FN_6**	$\sum_i FN_i$
Pityriasis rosea (4)	**TN**	**TN**	**FP_4**	**TN**	**TN**	**TN**	
Chronic dermatitis (5)	**TN**	**TN**	**FP_5**	**TN**	**TN**	**TN**	
Pityriasis rubra pilaris (6)	**TN**	**TN**	**FP_6**	**TN**	**TN**	**TN**	
			$\sum_i FP_i$				

Sensitivity evaluates the model ability to correctly classify the true sample and detect the presence of the disease. On the other hand, specificity measures the model ability to correctly reject other cases. The F1-score may be used as a single performance metric. It is the harmonic mean of sensitivity and specificity.

$$F1 = 2 \times \frac{Sensitivit\ \ y \times Specificit\ \ y}{Sensitivit\ \ y + Specificit\ \ y} \tag{14}$$

The seven performance measures of the four classification models have been computed for the testing data subset and the results have been tabulated in Table III. Fig. 7 plots the resulting performance metrics of the six dermatological classes.

TABLE III PERFORMANCE METRICS FOR INDIVIDUAL ALGORITHM ON THE CLASSIFICATION OF TEST SUBSET

	Class	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR	F1
CHAID	Psoriasis	97.3%	90.9%	100.0%	100.0%	0.0%	9.1%	95.2%
	seboreic dermatitis	94.7%	90.9%	95.6%	83.3%	4.4%	9.1%	87.0%
	lichen planus	98.2%	90.9%	100.0%	100.0%	0.0%	9.1%	95.2%
	pityriasis rosea	96.5%	76.9%	99.0%	90.9%	1.0%	23.1%	83.3%
	cronic dermatitis	94.7%	94.4%	94.7%	77.3%	5.3%	5.6%	85.0%
	pityriasis rubra pilaris	99.1%	100.0%	99.1%	83.3%	0.9%	0.0%	90.9%
Bagging	Psoriasis	98.2%	100.0%	97.6%	93.8%	2.4%	0.0%	96.8%
	seboreic dermatitis	97.3%	90.9%	98.9%	95.2%	1.1%	9.1%	93.0%
	lichen planus	99.1%	95.5%	100.0%	100.0%	0.0%	4.5%	97.7%
	pityriasis rosea	98.2%	92.3%	99.0%	92.3%	1.0%	7.7%	92.3%
	cronic dermatitis	100.0%	100.0%	100.0%	100.0%	0.0%	0.0%	100.0%
	pityriasis rubra pilaris	100.0%	100.0%	100.0%	100.0%	0.0%	0.0%	100.0%
Boosting	Psoriasis	97.3%	90.9%	100.0%	100.0%	0.0%	9.1%	95.2%
	seboreic dermatitis	95.6%	95.5%	95.6%	84.0%	4.4%	4.5%	89.4%
	lichen planus	99.1%	95.5%	100.0%	100.0%	0.0%	4.5%	97.7%
	pityriasis rosea	96.5%	76.9%	99.0%	90.9%	1.0%	23.1%	83.3%
	cronic dermatitis	97.3%	100.0%	96.8%	85.7%	3.2%	0.0%	92.3%
	pityriasis rubra pilaris	100.0%	100.0%	100.0%	100.0%	0.0%	0.0%	100.0%
NNET	Psoriasis	98.2%	93.8%	100.0%	100.0%	0.0%	6.3%	96.8%
	seboreic dermatitis	94.7%	95.5%	94.5%	80.8%	5.5%	4.5%	87.5%
	lichen planus	98.2%	90.9%	100.0%	100.0%	0.0%	9.1%	95.2%
	pityriasis rosea	97.3%	84.6%	99.0%	91.7%	1.0%	15.4%	88.0%
	cronic dermatitis	99.1%	100.0%	98.9%	94.7%	1.1%	0.0%	97.3%
	pityriasis rubra pilaris	100.0%	100.0%	100.0%	100.0%	0.0%	0.0%	100.0%

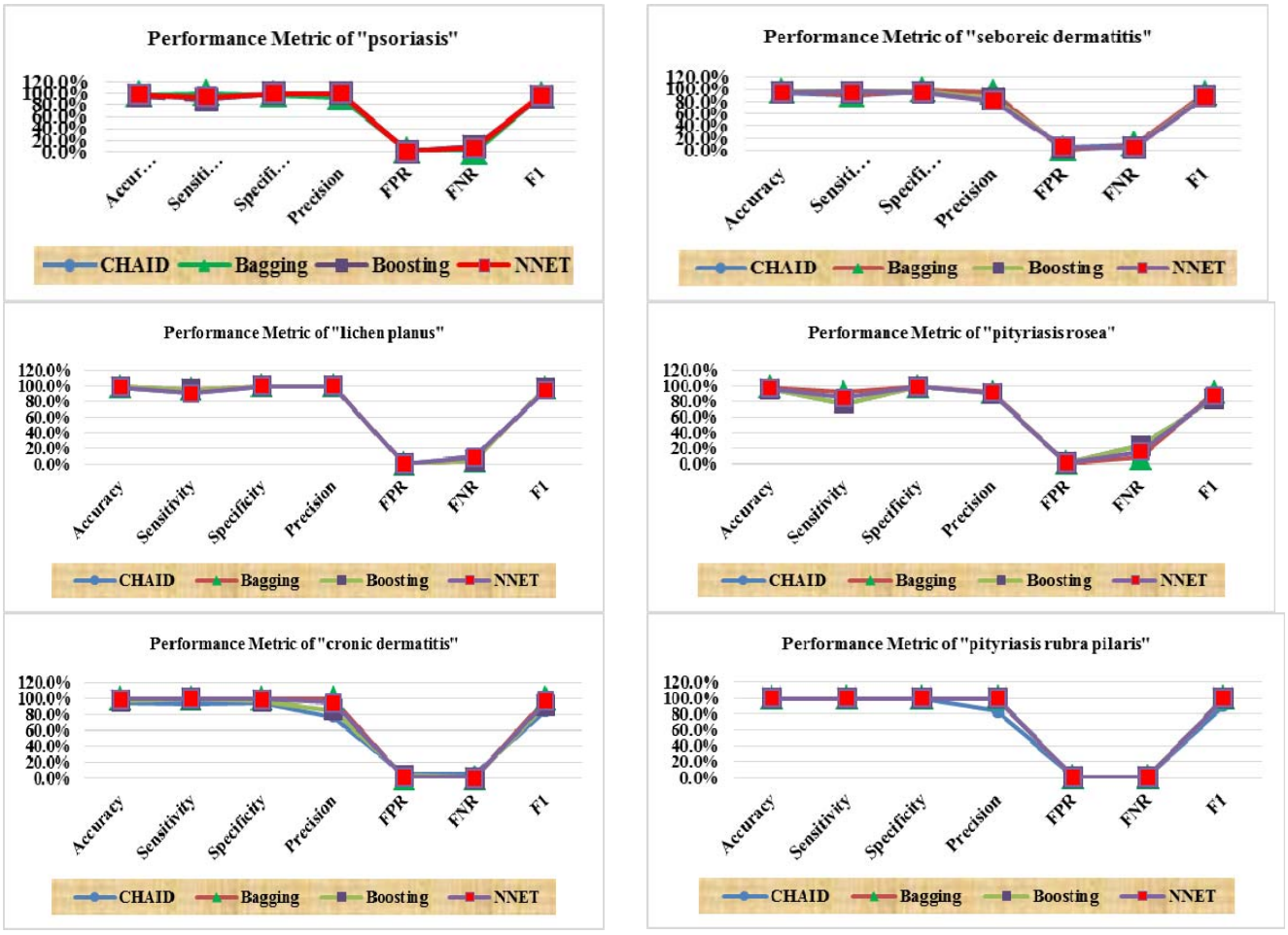


Fig. 7. Comparisons of the performances of CHAID, Bagging, Boosting and MLP NNET using different metrics on the classification of test subset

The original dataset is partitioned into training and testing subsets with the ratio 70:30, respectively. Training subset has been used during the model construction phase, while testing subset has been sued in the model evaluation phase. The objective is to measure the generalization and the power of the trained models. Using the total classification accuracy of testing subset as a single criterion for evaluating the strength of the modeling algorithm, the bagging ensemble achieved the first place among other models with accuracy of 96.46%. Both CHAID, boosting and NNET achieved good training accuracies but they are getting worse when classifying testing subset. Their training accuracies are 95.92%, 100% and 98.37% and their testing accuracies are 90.27%, 92.92% and 93.81% respectively. Experimental results in Table III and Fig. 7 show that bagging has slightly better performance than single CHAID, boosting (AdaBoost) or ANN with two hidden layers. Although, the CHAID decision tree suffers from the instability problem like other single decision tree model, it has a comparable performance to other modeling algorithms. It can be used when the objective is to build a transparent model to interpret the decision rather than to predict it. Table IV shows the agreement between CHAID and other models. It is noted that the ratio of agreement between CHAID and Boosting on the classification of testing subset is more than that with bagging or NNET.

In general, these results assure the potential benefit of the application of machine learning algorithms to analyze and diagnose the medical data. Bagging are considered stable modeling algorithms, that is, it is little affected by any little variations in the training subset. On the other side, boosting-based ensemble is a strong model, which improves the classification accuracy; despite, it has an overfitting problem.

TABLE IV AGREEMENT BETWEEN CHAID AND OTHER MODELS

Algorithm	Agreement	Testing	Training
Bagging	Agree	90.27%	97.14%
	Disagree	9.73%	2.86%
Boosting	Agree	93.81%	95.92%
	Disagree	6.19%	4.08%
NNET	Agree	89.38%	95.92%
	Disagree	10.62%	4.08%

V. CONCLUSIONS

This study applied single CHAID decision tree, CHAID-based bagging and CHAID-based boosting algorithms for the differential diagnosis of ESDs (skin diseases). Their experimental results have been benchmarked with those obtained from ANN. The dataset contains dermatological proven diseases' features obtained from the UCI machine learning repository. The dataset includes six groups of ESDs that share some signs

and symptoms with redness (erythema) causing losses in cells (squamous).

Experimental results showed that bagging has achieved a good and stable performance while boosting suffers from an overfitting problem. The single CHAID decision tree model has achieved a comparable performance. A single decision tree is an easy to use, very flexible and transparent. The CHAID algorithm uses multiway splits and not binary ones. It can generate wider tree and leads to better interpretation. However, it is an unstable algorithm; any little variations in the training data leads to different model structure. Bagging and boosting ensembles solve this instability problem, but on the expense of interpretability. The proposed stream corrects the distribution to the training subset to become more balanced. The study included detailed results for all experiments, tables and drawings used in different comparisons. The comparisons were based on seven different performance metrics, which are derived from the multiclass confusion matrix. They are the accuracy, sensitivity, specificity, precision, false positive rate, false negative rate, F1 score. Definitely, the tree-based ensembles; bagging and boosting have achieved better classification accuracy but they lose the transparency and interpretability of a single tree. The proposed future research work is to reduce the number of trees in the ensembles as much as possible to obtain the best classification accuracy and at the same time keep the transparency nature of a single tree.

REFERENCES

[1] H. A. Güvenira and N. Emeksiz, "An expert system for the differential diagnosis of erythemato-squamous diseases," *Expert Systems with Applications,* vol. 18, pp. 43-49, 2000.

[2] H. A. Güvenira, G. Demiröza, and N. İlterb, "Learning differential diagnosis of erythemato-squamous diseases using voting feature intervals," *Artificial Intelligence in Medicine,* vol. 13, pp. 147-165, 1998.

[3] M. E. B. Menai, "Random forests for automatic differential diagnosis of erythematoâ€“squamous diseases," *International Journal of Medical Engineering and Informatics,* vol. 7, pp. 124-141, 2017/11/25 2015.

[4] V. Kecman and M. Kikec, "Erythemato-Squamous Diseases Diagnosis by Support Vector Machines and RBF NN," in *Artificial Intelligence and Soft Computing: 10th International Conference, ICAISC 2010, Zakopane, Poland, June 13-17, 2010, Part I* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 613-620.

[5] M. J. Abdi and D. Giveki, "Automatic detection of erythemato-squamous diseases using PSOâ€“SVM based on association rules," *Engineering Applications of Artificial Intelligence,* vol. 26, pp. 603-608, 2013.

[6] L. Wang and T. Z. Sui, "Application of Data Mining Technology Based on Neural Network in the Engineering," in *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, 2007, pp. 5544-5547.

[7] R. Setiono, "Extracting Rules from Neural Networks by Pruning and Hidden-Unit Splitting," *Neural Computation,* vol. 9, pp. 205-225, 2017/11/25 1997.

[8] N. Lin, D. Noe, and X. He, "Tree-Based Methods and Their Applications," in *Springer Handbook of Engineering Statistics*, H. Pham, Ed. London: Springer London, 2006, pp. 551-570.

[9] G. V. Kass, "An Exploratory Technique for Investigating Large Quantities of Categorical Data," *Journal of the Royal Statistical Society. Series C (Applied Statistics),* vol. 29, pp. 119-127, 1980.

[10] D. Biggs, B. De Ville, and E. Suen, "A method of choosing multiway partitions for classification and decision trees," *Journal of Applied Statistics,* vol. 18, pp. 49-62, 1991.

[11] M. Lichman, "UCI Machine Learning Repository," 2013.

[12] Y. Wang and J. Xie, "Granular Computing Combined with Support Vector Machines for Diagnosing Erythemato-Squamous Diseases," in *Health Information Science: 6th International Conference, HIS 2017, Moscow, Russia, October 7-9, 2017, Proceedings* Cham: Springer International Publishing, 2017, pp. 56-68.

[13] H. Liu, J. Sun, L. Liu, and H. Zhang, "Feature selection with dynamic mutual information," *Pattern Recognition,* vol. 42, pp. 1330-1339, 2009.

[14] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*: Taylor & Francis, 1984.

[15] J. H. Friedman, "Multivariate Adaptive Regression Splines," *Ann. Statist.,* vol. 19, pp. 1-67, 1991.

[16] S. L. Salzberg, "C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993," *Machine Learning,* vol. 16, pp. 235-240, 1994.

[17] J. R. Quinlan, "C5.0 decision tree software, available at http://www.rulequest.com," 1999

[18] G.-W. Cha, Y.-C. Kim, H. J. Moon, and W.-H. Hong, "New approach for forecasting demolition waste generation using chi-squared automatic interaction detection (CHAID) method," *Journal of Cleaner Production,* vol. 168, pp. 375-385, 2017.

[19] M. Abdar, M. Zomorodi-Moghadam, R. Das, and I. H. Ting, "Performance analysis of classification algorithms on early detection of liver disease," *Expert Systems with Applications,* vol. 67, pp. 239-251, 2017.

[20] M. Tayefi, H. Esmaeili, M. Saberi Karimian, A. Amirabadi Zadeh, M. Ebrahimi, M. Safarian, M. Nematy, S. M. R. Parizadeh, G. A. Ferns, and M. Ghayour-Mobarhan, "The application of a decision tree to establish the parameters associated with hypertension," *Computer Methods and Programs in Biomedicine,* vol. 139, pp. 83-91, 2017/11/26 2017.

[21] K. Y. Chung, S. Y. Oh, S. S. Kim, and S. Y. Han, "Three representative market segmentation methodologies for hotel guest room customers," *Tourism Management,* vol. 25, pp. 429-441, 2004.

[22] D. Merel van and F. Philip Hans, "Evaluating chi-squared automatic interaction detection," *Inf. Syst.,* vol. 31, pp. 814-831, 2006.

[23] S. R. Swarnalatha and K. N. G.M., "Analysis of Optimization Techniques in Chi-Squared Automatic Interaction Detection," *Journal of Theoretical and Applied Information Technology,* vol. 65, pp. 813-823, 2014.

[24] "IBM SPSS Modeler 18.0 Algorithms Guide," 2016.

[25] G. Paleologo, A. Elisseeff, and G. Antonini, "Subagging for credit scoring models," *European Journal of Operational Research,* vol. 201, pp. 490-499, 2010.

[26] W. Gang, H. Jinxing, M. Jian, and J. Hongbing, "A comparative assessment of ensemble learning for credit scoring," *Expert Syst. Appl.,* vol. 38, pp. 223-230, 2011.

[27] R. E. Schapire, "The strength of weak learnability," *Machine Learning,* vol. 5, pp. 197-227, 1990.

[28] F. Yoav and E. S. Robert, "Experiments with a new boosting algorithm," in *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning* Bari, Italy: Morgan Kaufmann Publishers Inc., 1996, pp. 148-156.

[29] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning, Data Mining, Inference, and Prediction, Second Editio*. Stanford, California: Springer-Verlag New York, 2009.

[30] V. L. Berardi and G. P. Zhang, "The Effect of Misclassification Costs on Neural Network Classifiers," *Decision Sciences,* vol. 30, pp. 659-682, 1999.

[31] F. Neukart, "An outline of artificial neural networks," in *Reverse Engineering the Mind: Consciously Acting Machines and Accelerated Evolution* Wiesbaden: Springer Fachmedien Wiesbaden, 2017, pp. 39-101.

[32] Miriam Rodrigues Silvestre and Lee Luan Ling, "Pruning methods to MLP neural networks considering proportional apparent error rate for classification problems with unbalanced data," Measurement, vol 56, pp 88-94, 2014.

[33] K. Bartosz, Micha, Woniak, and S. Gerald, "Cost-sensitive decision tree ensembles for effective imbalanced classification," *Appl. Soft Comput.,* vol. 14, pp. 554-562, 2014.

[34] M. L. McHugh, "The Chi-square test of independence," *Biochemia Medica,* vol. 23, pp. 143-149, 2013.

[35] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters,* vol. 27, pp. 861-874, 2006.