RBDT: The Cascading of Machine Learning Classifiers for Anomaly Detection with Case Study of Two Datasets

Goverdhan Reddy Jidiga and Porika Sammulal

Abstract. The inhuman cause of behavior in computer users, lack of coding skills pursue a malfunctioning of applications creating security breaches and vulnerable to every use of online transaction today. The anomaly detection is in-sighted into security of information in early stage of 1980, but still we have potential abnormalities in real time critical applications and unable to model online, real world behavior. The anomalies are pinpointed by conventional algorithms was very poor and false positive rate (FPR) is increased. So, in this context better use the adorned machine learning techniques to improve the performance of an anomaly detection system (ADS). In this paper we have given a new classifier called rule based decision tree (RBDT), it is a cascading of C4.5 and Naïve Bayes use the conjunction of C4.5 and Naïve Bayes rules towards a new machine learning classifier to ensure that to improve in results. Here two case studies used in experimental work, one taken from UCI machine learning repository and other one is real bank dataset, finally comparison analysis is given by applying datasets to the decision trees (ID3, CHAID, C4.5, Improved C4.5, C4.5 Rule), Neural Networks, Naïve Bayes and RBDT.

Keywords: Anomaly detection, C4.5, Decision tree, Naïve bayes, RBDT.

1 Introduction

Anomaly detection [2] is a kind of intrusion detection to model the behavioral patterns in image and medical applications, novelties in industrial machine malfunctions, fraud accounting actions in financial banking sectors and also the

Goverdhan Reddy Jidiga
Department of Technical Education, Govt. of Andhrapradesh, Hyderabad, India
e-mail: jgreddymtech@gmail.com

Porika Sammulal
JNTUH College of Engineering, Karimnager, JNTU University, Hyderabad, India
e-mail: sammulalporika@gmail.com

El-Sayed M. El-Alfy et al. (eds.), *Advances in Intelligent Informatics*,
Advances in Intelligent Systems and Computing 320, DOI: 10.1007/978-3-319-11218-3_29

anomalies in all kinds of network applications. Today the intrusion detection systems (IDS) [1] are modeled by the various conventional, adorned machine learning approaches (or classifiers) and Meta classifiers. The anomalies have various dimensions and detecting them as false is depending on the ratio of dynamic, online issues to be considered in model. As on today the world is moving in competitive directions and no one is follow the ethical values. Like in medical diagnosis, anomaly boundaries need to model into 3-layer security [8], it is help to focus on the awareness and required to reduce the design cost of algorithms in information security. The multilevel classification in adorned machine learning is more decent method to identify and model the zero day attacks in latest critical infrastructure applications.

1.1 Machine Learning in Anomaly Detection

The Anomaly based intrusion detection system [2, 3] is a system for detecting computer intrusions and anomalous behavior by monitoring system activity, incoming and outgoing internet traffic based on applications and classifying it as either normal or anomalous. In this paper the Anomaly detection system (ADS) is a module to detect the abnormal samples (records) or anomalies by monitoring decision tree transition and categorizes them as either regular behavior or anomalous through observing class label. In this security field many IDSs [2,3,5,7] techniques have been developed for detecting and modeling anomalies in structured and unstructured multi dimensional data in traditional, real time and critical infrastructure applications, but still impossible to catch all latest anomalies. So machine learning is useful in this area to achieve satisfactory results in ADS. Machine learning techniques (predict known) [6,7] are generally different from data mining (predict unknown)[4] and facilitate users to extract new features from datasets and construct a new system to solve predictive issues like novelties, anomalies, outliers. The detection of all done by anomaly detection by imposing classification, ranking of features, learning control and outcomes decision analysis. Hence the machine learning in anomaly detection [5] always enables to automate the system massively from big database sources by constructing novel rules using mathematical hardness in real time critical infrastructure applications and improve the results. In this paper the outline of the work is as follows, in section-2 we have given huge literature review, section-3 gives the framework and algorithms of proposed work, sectin-4 briefly explains about case studies, section-5 confer experiments with results and finally discussions, future work given.

2 Related Work

The decision tree (DT) [35] is powerful learning classifier used in anomaly detection [7] and today it is completely designed in terms of dynamic rule sets extracted from learning. Initially the popular classification algorithm ID3 [10] designed based on heuristics and it is a top down, divide and conquer, greedy based and

entropy-gain technique. The C4.5 [11-13] is a posterior technique for ID3 and used still now in many applications successfully giving optimal solutions by change of some parameters. There are many classification techniques available in machine learning like normal rule based classifiers , Naïve Bayes(NB)[34] , Support vector machines (SVM), Neural networks (NN)[31] and DT based algorithms[35] CLS, GUIDE, QUEST, CHAID[9], CART[16], C5.0 and multivariate decision trees. All these algorithms have advantages and disadvantages when use in ADS given in this paper as follows.

The QUEST [24] is based on univariate attribute domain using ANOVA F-test, but it uses 10-fold cross validation in training is only positive here. The CHAID [9] designed to handle nominal attributes based on p-value, chi-square test and likelihood ratio. The CHAID tree is an attractive DT over its precedents like AID, THAID and its construction begins with complete data space and continues based on repeated and homogeneous subspaces into two or more child nodes. But this algorithm treats the missing instances as same unit of single and also lack in pruning. CART [16, 23] is a recursive method used in regression and also good for classification, but the split is limited to only two. It uses cost complexity pruning takes additional cost.

The popular C4.5 [11-14] introduced with a many new features and it constructs a tree by considering all attributes get equal priority assume most significant and optimize the decision rule by well pruning. C4.5 also suffering with null instances, overfitting and insignificant attributes in some cases. This is still good due to efficiency of algorithm and adjusts with other bagging and boosting techniques. In [18] they used 'one against all approach' with C4.5 on three sets of UCI datasets, got good estimation and in [19] uses this for outlier detection by hybrid process, but they have not given correct results with their data, Later in [22] uses hierarchical clustering with same data shown well. In [21] uses this C4.5 for remote sensing data along rough sets also got good results. Finally C4.5's successor C5.0 shown some improvements to previous, but not well due to heard to learn compare to learning of C4.5 generally fast [14, 20]. C4.5 Rule [14, 15, 30] is rule based induction tree and it is uses pruning heuristics to improve the accuracy by remedial strategy of derive, generalize, group and ordering the rules. But the problem with continuous value attribute is rules are needed to update continuously. Other rule based CN2 presented in [25], evaluates possible conjunction of attribute tests conditions of a rule based on I_g (entropy) measure. Later in [26], the measure is changed to get accuracy more, but frequent change in rules then training is a complicated in learning take many iterations for rule search. The STAGGER, FLORA3, AQ-PM also kind of rule learning systems uses numerical data. Some uses decision rules by overlapping and in all above adapting new rules and remove old is drift in accuracy also. Improved C4.5 [27, 29] is a successor of C4.5 uses generalized entropy with new parameter β and improved gain ratio used instead of I_g (standard). Later cost sensitive C4.5 [28] (version of C4.5) uses misclassification costs matrix in training process, in [30] Enhanced C4.5 algorithm for intrusion detection in networks by extracting a set of classification rules from KDD dataset.

Neural networks [31] used in ADS with different approaches and like multi layer perceptrons (MLP) is popular to classify the data with threshold, activation function and error adjustment to improve the accuracy. Naïve bayes (NB) [34] is simple and better predictive accuracy over ID3 and also refining NB classifier [42] is good. C4.5 in some well organized data without any discrepancies; it is extremely efficient like ensemble classifiers (bagging and boosting) to combine classifier predictions under the Gaussian naive bayes (GNB) assumptions. AdaBoost [6] algorithm is developed with incremental refinement over large datasets to classify the data instances and they are finalized the boosting with k-NN is better than bagging with C4.5 in terms of accuracy. Random Forest [6, 16] is again shows the performance is better than AdaBoost. Finally the latest work in this field is structure learner [41], uses the features of DT and mapped into markov network structures to improve the learning very fast and even for complex data instances. Other extensions of DTs are like oblivious decision tree, RBDT-1[39], neurotree [40] (light weight IDS) and fuzzy decision tree also useful in ADS, but in case large datasets parallel ID3, SLIQ, SPRINT like useful to get good results. In [38, 43], uses decision semantics to remove the rules based on irrelevant conditions in tree over the process of converting the rules.

3 Proposed Work: RBDT

Decision tree (DT) [35] is a kind of machine learning [5] algorithm to solve the classification problems and make simplify the solution with genuine performance parameters like predictive accuracy (maximize) and false positive rate (minimize).

The RBDT (Rule Based Decision Tree) is multi-level classification model. It is a combination or a kind of cascading of Naïve bayes and C4.5 rules used to classify the on-the-fly data by framing dynamic rules from existing classifier including with new behavior. In RBDT the rules are not predictable, but dynamic while learning. The decision tree is addressing the frequent anomalies by calculating performance parameters associated with ADS at every node which is label the class C as leaf. The RBDT algorithm also support multi-way split which has more than two out comes at each node. The RBDT is constructed for the given datasets is shown in Fig.1 (bottom). Here the attribute PAN is selected as a root due to high information gain (I_g) value compared to the candidate attribute set has {PAN, P-MODE, DATE}. The RBDT is constructed in this paper is shown partially in Fig.1, but full tree is very complex and invalid records treated as error data in this dataset-2.

3.1 Frame Work of RBDT

The frame work of RBDT is given in Fig.1, it is a process begin with applying the Gaussian naïve bayes (GNB) on selected datasets as input and generation of Gaussian segments $S = \{S_0, S_1, S_2 \ldots S_n\}$like criteria used ten-fold cross validation

in training of sample data. The rules are generated from each segment separately taken into one set (α-rules) and collectively from all segments (original dataset) create another set (β-rules). In this the rules are generate from applying C4.5 rules algorithm and form a pair of rule set $\{C_\alpha, C_\beta\}$ and same as with naïve bayes (NB) form a rule set $\{N_\alpha , N_\beta\}$ shown in Fig.2 .The GNB is based on naïve bayes algorithm used to classify the data or samples, but in this case we have used it as pre-classification to observe the analysis of training help to improve the performance in large datasets.

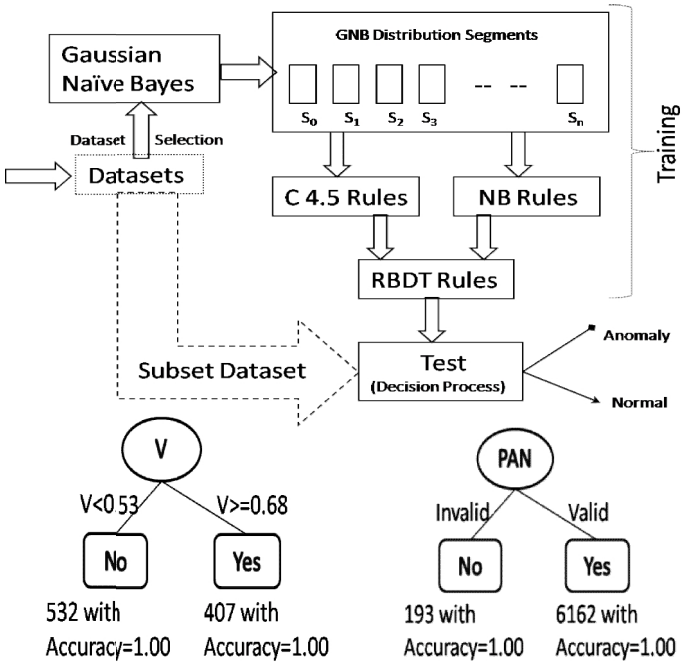

Fig. 1 The frame work of RBDT has RBDT rules $\{ R_\alpha , R_\beta \}$conjunction of NB$\{N_\alpha , N_\beta\}$, C4.5$\{C_\alpha, C_\beta\}$ Rules (top) and construct the decision tree based on RBDT rules and continue the test process to determine the outcome of new data as input. In Fig (bottom left) shows the simple RBDT for dataset-1(Banknote authentication) and for training set of dataset-2 (real-time bank dataset), the tree constructed on attribute {PAN} based on high I_g by PAN (bottom right).

We have selected GNB only, because it supports for continuous data type well compared to Gaussian based Bernoulli and multinomial models used in discrete data as input. In this the dataset (X) has instances$\{X_0, X_1, \ldots X_n\}$ of continuous type segmenting into S_i by computing a mean, variance; co-variance is given by following GNB instead of only two segments of class-yes and class-no.

Rule #	Rule Description	Class (ACC)
Rule-1	V>=0.26&V<0.53&S>=5.85	No(0.98)
Rule-2	(V>=0.53&E>=0.28)&(V>=0.63&C>=-4.95)	No(0.99)
Rule-3	(V>=0.53&E>=0.28)&(V<0.63&C<-2.25&S>=5.95)	No(0.99)
Rule-4	(V>=0.53&E>=0.28)&(V<0.63&C>=-2.25&E<0.79)	No(0.99)
Rule-5	(V>=0.53&E>=0.28)&(V<0.63&C>=-2.25)&(E>=0.79&C>=1.85)	No(1.00)
Rule-6	(V>=0.53&E>=0.28)&(V<0.63&C>=-2.25)&(E>=0.79&C<1.85&S>=3.85)	No(1.00)
Rule-7	(V<0.53&S<5.85)&(C>=3.5&S>=-1.85&V>=0.35)	No(1.00)
Rule-8	V<0.53&(S>=5.85&V<0.26)	Yes(0.96)
Rule-9	(V>=0.53&E>=0.28)&(V<0.63&C<-2.25&S<5.95)	Yes(1.00)
Rule-10	V<0.53&S<5.85&C<3.05	Yes(1.00)
Rule-11	(V<0.53&S<5.85)&(C>=3.05&S<-1.85&V<0.46)	Yes(0.99)

Rule #	Rule Description	No. of Instances	Class (ACC)
Rule-1	V>=0.68	393	No(1.00)
Rule-2	(V>=0.53&E>=0.28)&(C>=-4.95)	484	No(0.99)
Rule-3	(S>=5.15&S<=9.65)&(C<-4.45&(<8.85)	192	No(0.72)
Rule-4	V<0.53	305	Yes(1.00)
Rule-5	(V<0.53&S<5.85)&C<3.05	305	Yes(1.00)
Rule-6	(V<0.53&S<5.85)&(C>=3.05&S<-1.85)	181	Yes(1.00)

Rule #	Rule Description	No. of Instances	Class (ACC)
Rule-1	(V>=0.31&E>=0.28)&(C>=4.95)	484	No(0.99)
Rule-2	V>=0.68	407	No(1.00)
Rule-3	V<0.53	532	Yes(1.00)
Rule-4	(V<0.31&S<5.85)&C<3.05	305	Yes(1.00)
Rule-5	(V<0.31&S<5.85)&(C>=3.05)	181	Yes(1.00)

Fig. 2 C4.5 rules (top), NB rules (bottom left) and RBDT rules (bottom right) for Banknote authentication, here V-variance, E-entropy, C-kurtosis, S-skewness, class labels(Yes/No) and ACC- accuracy.

$$P\left(x = \frac{v}{c}\right) = \frac{1}{\sqrt{2\pi\sigma_c^2}} . -e^{\frac{(v-\mu_c)^{2\cdot}}{2\sigma_c^2}} \quad (1)$$

Where x-continuous attribute, μ_c -mean value of x of class c for each segment, σ - is variance of x of class, P is a probability density of some value in segment for a given class C. v- a instance and new instance (in test) to determine its class.

The rules collected from GNB segments shown in Fig.2. Here the rules are given for collective segments maximum and very less at individual segment. The rules which give accuracy high only shown in Fig.2 and remaining rules having less support and less accuracy.

3.2 Issues in Decision Trees

Overfitting: Overfitting is problem of decision tree create training set error and reduce the accuracy by poor criteria and selection of rules. Generally this can avoided by general pruning techniques like cross validation (10-fold), cost complexity pruning[16], reduced error pruning [11,12], minimum error pruning,

pessimistic pruning, error based pruning [14], optimal pruning. All these are maximum use the bottom up approach to get good performance. Optimal pruning with Δ is good for all.

Pruning: The Tree pruning [17] is a property of decision trees to avoid the overfitting problems and if any sub tree is unlikely to make less accuracy then possibility to prune the tree at any level to increase accuracy until satisfactory performance. In these case many techniques available based on pre and post pruning. The pruning cost is depending on pruning criteria and technique. If pruning is done at probability of node(v) , which has a set of positives (p) and negatives(n) with expected count of irrelevant P_k and N_k in each subsets P_s and N_s. If v=n+p, then

$$P_k = p * \frac{P_k+N_k}{p+n}, \quad N_k = n * \frac{P_k+N_k}{p+n} \tag{2}$$

$$\Delta = \sum_{k=1}^{d} \frac{(P_s-P_k)^2}{P_k} + \frac{(N_s-N_k)^2}{N_k} \tag{3}$$

Δ is distributed according to $\chi 2$ and take decision to accept or not. It will determine the cost of pruning if tree grows in unordered by describing error or noise.

3.3 Algorithms

In this paper, our work flow is divided into two algorithms one is pre-classification and other is generating decision tree. In algorithm-1, how the RBDT framework to be carried out in terms of pseudo code statements.

Algorithm-1: RBDT Pre-Classification
Input: *Samples Set (S), Gaussian parameters,*
Output: *Rule_set (R)*

```
(1). Prepare dataset;
(2). Do Normalization of data;
(3). Apply GNB classifier and segmenting the dataset;
(4). Apply C4.5 rule algorithm and Naïve Bayes algorithm;
(5). Prepare RBDT rules Rule_set(R);
(6). Call generate_RBDT;
```

In algorithm-2, the RBDT construction is explained for only two childs and shown one of RBDT for dataset-2 in Fig.1 (right). O_i is depends upon creation of no. of childs at L_i for A_i. For this the dataset (sample-set) has proper attribute domain (AD) represented as AD (A_n U C_n), here A denote attribute set contains 'n' attributes then A= {A_1, A_2 A_3 …A_n} and C is a class label {yes, no}. In training the set of instances (samples or records) are described by T(s) = {(x_1, y_1), (x_2, y_2) … (x_n, y_n)}, here x_iЄX and X is data value (except class) is determined by attribute domain and y is a class instance of domain Y. The generalized classification is simply done by y=f(x), f is a function determined by rules.

Algorithm-2: Generate_RBDT

Input: *Sample_Set (S), attribute_list (A), Key_Attribute, Rule_set(R) and Candidate_attribute_set (K);*

Output: *RBDT*

(1).Create node N;
If samples of dataset is same class;
Then return N as a leaf node and labeled with Class C;
(2).If $A_i \in K_i$
Then calculate information-gain (I) for each attribute A_i*;*
(3).If $I(A_i)>I(A_j)$ *then* A_i *is a test attribute as Root; Else* A_j*;*
(4).Create L_i *and* R_i *to Root by applying rule* $R_i \in R[R_1 \ldots R_n]$*;*
(5).Divide the samples into sub-roots L_i *and* R_i *;*
If L_i *or* R_i *not generated then apply* R_{i+1} *and goto step (4);*
(6).For each sub-root compute optimality cost O_i*;*
If O_i (A_i) *at Level* $L_i \neq O_i$ (A_i) *at Level* L_{i+1}*;*
Then goto step (7); else PRUNE (T, t_i*) at Level* L_{i+1} *;*
(7).For each sub-root, if not Labeled, $A \neq NULL$, $S \neq NULL$
Goto step (5) else Labeled as class C;
(8)Generate_RBDT(S, A);

The information gain is calculated by following steps and the Entropy (H) is provide the splitting criteria variable used to find the purity (or impurity) of attribute.

$$\mathrm{H}\left(\frac{\mathrm{p}}{\mathrm{p+n}},\frac{\mathrm{n}}{\mathrm{p+n}}\right) = -\frac{\mathrm{p}}{\mathrm{p+n}}\log_2\frac{\mathrm{p}}{\mathrm{p+n}} - \frac{\mathrm{n}}{\mathrm{p+n}}\log_2\frac{\mathrm{n}}{\mathrm{p+n}} \tag{4}$$

Here p-the probability of positives (yes or true samples), q-the probability of negatives (no or false samples).Generally the training set has combination of p positives and n negatives and which are input to root to train. In this the attribute variance (in dataset-1) and PAN (in dataset-2) contains different values so the expected entropy is required, so the expected entropy (EH) (Average Entropy of children's) is calculated for attribute A as:

$$\mathrm{EH(A)} = \sum_{\mathrm{i=1}}^{\mathrm{k}} \frac{\mathrm{p_i+n_i}}{\mathrm{p+n}} \cdot \mathrm{H}\left(\frac{\mathrm{p_i}}{\mathrm{p_i+n_i}},\frac{\mathrm{n_i}}{\mathrm{p_i+n_i}}\right) \tag{5}$$

Where the k-distinct value or partitions, i- particular child and $p_i + n_i$ is assume as parent then information-gain (A) is:

$$\mathrm{I(A)} = \mathrm{H}\left(\frac{\mathrm{p}}{\mathrm{p+n}},\frac{\mathrm{n}}{\mathrm{p+n}}\right) - \mathrm{EH(A)} \tag{6}$$

For dataset-2, the decision tree constructed based on the attribute candidate set C has three attributes PAN, P-MODE, DATE and for all these then I_g is calculated, but attribute PAN has I_g and variance attribute has highest information-gain for dataset-1.

4 Case Studies

4.1 Datasets

For our experiment, we have chosen two datasets 1) Banknote Authentication from UCI machine learning repository and 2) Real time bank dataset collected from private CA (Chartered Accountant).

Banknote authentication: This dataset provided at UCI machine learning repository [36] and it is collection of feature data extracted from image, transformed into 4 attributes of multivariate data. This dataset is actually extracted to authenticate the bank specimens and the data was digitized by 400x400 pixel ratio with 600 dpi resolution. This dataset has 1372 observations (instances) without any missing values with 4 continuous and 1 class attribute. For our experiment the classification task is trained with 500 observations and tested on remaining samples with yes or no (1 or 0) basis.

Real time bank dataset: The dataset is a bank database consists of 51095 records about transactions made during 2006 and 2013, but in that maximum transactions are held in 2008 to 2010. Out of these 6356 unique records are identified and consider as training set with learning rules .The original data set has 9 attributes, but we have concentrated on attributes which has highest $I_g(E)$. We have given the experimental statistics on this dataset and 3 core attributes was considered as key attribute set to elevate results in work.

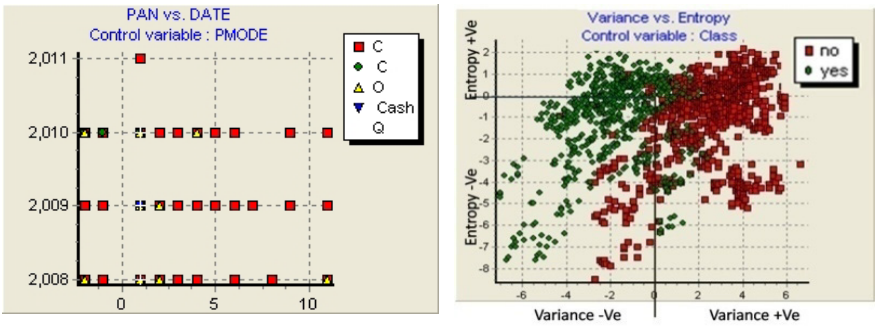

Fig. 3 Scatter plots for two datasets. Dataset-2 is depending on control variable as PMODE (left) shows how many records are grouping into bins in each year, here {Cash, C,C} are same kind, Q-cheque, O-other. In Fig (right) shows on dataset-1 control variable as class and the regions determine negative instances of entropy and variance.

4.2 Normalization of Datasets

Banknote authentication: This dataset has two main key attributes which are Variance and Entropy has high I_g, but as per statistical ethics the variance and entropy are a non-negative [37] and also mutually inversely proportional to each other and

plots shown in Fig.3 (right). In Gaussian (normal distribution) instance, the mutual relation and dependency is exist between variance and entropy: it is like maximizes entropy (E) for a given variance (V) and at the same time minimizes the variance for given entropy. In our experiment we need to fit the data in specific range and classify the samples effectively by different normalization criteria. We have chosen the function mat2grey () to give the range of values between 0 and 1. The mat2grey function is used to convert matrix in scaling of image values between 0 and 1

Real time bank dataset: The dataset has different combination of attributes and we know that the attribute PAN has highest I_g , but it is continuous and alphanumeric based value. So it is not possible to classify the records as per original set, for this we have uses groups of PAN status and substituted with decimal value and we have given some substitutions for PAN status given in Fig.3 (left). The PAN is split attribute and the next level attribute is PAN-Status is normalized by decimal scaling ($V/10^2$).

5 Experiments and Results

In this paper, for our experimental work, the datasets related to same application was taken and consider as benchmark. The both case studies (Banknote authentication from UCI [36] and Real time bank dataset) are simulated on Intel Pentium CPU 3GHz speed, 2GB RAM, Windows-XP OS , Matlab -32 bit version[32] for Rules and performance benchmark for Case studies is SIPINA data mining tool[33].

Parameters \ Alg'm	ID3			CHAID			C4.5			NN	NB	RBDT	
Confidence Level (P-Value) -parameters	0.05	0.05	0.07	0.05 0.01	0.07 0.01	0.07 0.01	25 2	50 2	75 2	1001 5 n/l	U.C.C.D	25 2	50 2
Sampling	500	1000	500	500	1000	500	500	500	1000	500	500	500	1000
Random Sample size (Filter)	50%	72%	50%	50%	50%	72%	50%	50%	72%	50%	50%	50%	72%
Idle Samples	686	385	686	686	686	385	686	686	385	676	676	676	385
Learning Time	0.32s	0.34s	0.33s	0.34s	0.34s	0.34s	0.37s	0.36s	0.37s	0.32s	0.12s	0.62s	0.69s
Testing Time	0.31s	3.44s	0.32s	0.37s	0.31s	0.32s	0.34s	0.32s	0.34s	0.38s	0.25s	0.52s	0.64s

Fig. 4 Describe the learning time and testing time with different kind of training parameters taken for different algorithms on dataset-1 (Banknote authentication). Here NB takes very less time compare to all, ID3 also take less time due to simplicity in algorithm, where as others CHAID, C4.5, NN takes more time due to additional parameters and rules play major role. In RBDT, take much more time.

In Fig.4 and Table.1, the algorithms like ID3, CHAID, C4.5 and RBDT are based confidence level (or p-value). For NN, multi layer perceptrons (MLP) considered with parameters are no. of iterations, nodes per layer, one hidden layer and maximum error =0.05. All parameters are taken at default initially and gradually

changed. In naïve bayes (NB) the default prior consider as unconditional class distribution (UCCD) and same as all classes (All C). In Table.1, we have shown all possible results with different combinations and highlighted some values showing good in performance.

Table 1 Shows the performance of all algorithms tested on dataset-1. Here FPR-false positive rate, DR-detection rate, ACC-accuracy, ER-error rate. All results were taken at training set of 500 samples.

Samples (Observations) Tested:	**50%**				**72%**				**100%**			
Method	**FPR**	**DR**	**ACC**	**E.R**	**FPR**	**DR**	**ACC**	**E.R**	**FPR**	**DR**	**ACC**	**E.R**
ID3 p = 0.05	0.05	0.90	0.93	0.07	0.05	0.89	0.92	0.08	0.07	0.92	0.92	0.08
ID3 p = 0.07	0.10	0.90	0.91	0.09	0.07	0.89	0.93	0.07	--	--	--	--
CHAID p = 0.05, 0.01	0.14	0.94	0.89	0.11	0.07	0.91	0.92	0.08	0.08	0.92	0.92	0.08
CHAID p = 0.07, 0.01	0.05	0.90	0.93	0.07	0.09	0.90	0.91	0.09	--	--	--	--
C4.5 25, 2	0.08	0.93	0.93	0.07	0.09	**0.96**	0.93	0.07	0.09	**0.95**	**0.93**	0.07
C4.5 15, 2	**0.01**	0.90	0.94	0.06	0.11	0.91	0.90	0.10	--	--	--	--
C4.5 50, 2	0.08	0.89	0.91	0.09	0.07	0.90	0.91	0.09	0.10	0.93	0.91	0.09
C4.5 75, 2	0.09	0.90	0.91	0.09	0.06	0.88	0.92	0.08	--	--	--	--
C4.5 75, 3	0.08	0.89	0.91	0.09	--	--	--	--	0.10	0.93	0.91	0.09
I-C4.5 25, 2	0.12	0.86	0.87	0.13	0.06	0.90	0.92	0.08	--	--	--	--
I-C4.5 15, 2	0.04	0.95	0.93	0.07	0.07	0.07	0.84	0.16	0.10	0.95	0.92	0.08
I-C4.550, 2	**0.02**	0.82	0.87	0.13	0.12	0.82	0.86	0.14	--	--	--	--
I-C4.5 75, 2	0.13	0.86	0.87	0.13	--	--	--	--	0.09	0.88	0.85	0.15
I-C4.5 75, 3	0.11	0.87	0.90	0.10	--	--	--	--	0.10	0.81	0.86	0.14
Rule C4.5 25	0.05	**0.96**	**0.94**	0.06	0.06	0.95	0.92	0.08	0.09	0.92	0.91	0.09
Rule C4.5 50	--	--	--	--	0.09	0.94	0.92	0.08	0.09	**0.99**	**0.99**	**0.01**
NN (MLP) 100, 5	**0.01**	0.91	0.96	0.04	**0.01**	0.62	0.72	0.28	**0.01**	0.92	0.96	0.04
NN (MLP) 200, 5	**0.01**	0.91	0.96	0.04	--	--	--	--	**0.01**	0.92	0.96	0.04
NB Prior=0.5 UCCI	0.06	0.92	0.93	0.07	0.08	0.95	0.93	0.07	0.07	**0.95**	**0.94**	0.06
NB Prior=0.5 All C	0.06	0.93	0.94	0.06	0.06	0.92	0.93	0.07	0.07	0.95	0.94	0.06
RBDT 25, 2	0.03	**0.98**	**0.97**	0.03	0.05	**0.96**	**0.96**	0.04	0.07	**0.96**	**0.98**	0.02
RBDT 50, 2	0.03	**0.97**	**0.98**	0.02	0.05	0.95	**0.96**	0.04	0.04	**0.97**	**0.97**	0.03

Parameters \ Alg'm	ID3			CHAID			C4.5			RBDT	
Confidence Level (P-Value) -parameters	0.05	0.05	0.1	0.05 0.01	0.05 0.02	0.1 0.01	25 2	50 2	25 2	25 2	50 >=2
Sampling	500	1000	5000	500	1000	5000	500	1000	6355	1000	5000
Random Sample size (Filter)	50%	75%	97%	50%	75%	97%	50%	75%	100%	75%	100%
Idle Samples	3177	1589	191	3178	1589	191	3178	1589	0	1589	0
Learning Time	0.94s	0.78s	0.63s	0.34s	0.78s	0.63s	0.62s	0.62s	0.62	0.82s	0.96s
Input Ratio	97% 0.03% 2.87%	97% 0.9% 2.3%	97% 0.03% 2.87%	97% 0.03% 2.87%	97% 1% 2%	97% 0% 3%	96% 1% 3%	97% 0% 3%	97% 0% 3%	96% 1% 3%	97% 0.03% 2.87%

Fig. 5 Describe the learning time and testing time with different kind of training parameters taken for different algorithms on dataset-2 (Real time bank dataset). Here we also shown input ratio of participating samples in training.

In Fig.5 and Table.2, the dataset-2 is simulated and results were given on default parameters and after little changes. For all the training parameters were consider same as in dataset-1, but compare to first the second case study showing poor performance. The accuracy is almost equal in both cases, but not in FPR. The some of the best results are highlighted for all algorithms applied on dataset. In this section we have shown the possible experimental work on both datasets with visual plots in Fig.6 and Fig.7. The plots are used here to present comparison of performance parameters. In Fig-6, we observe that our algorithm RBDT is well in predictive accuracy to classify the instances correctly, but for dataset-2 it is little bit low compares to C4.5 and ID3. The RBDT algorithm is also good for the indication of low FPR compare to all, in dataset-1 NN also give 0.1 only.

Table 2 Shows the performance of all algorithms tested on dataset-2. Here FPR-false positive rate, DR-detection rate, ACC-accuracy, ER-error rate. All the results were noted on training of 6356 records (data instances) and test on remaining.

Records Tested	50%				100%			
Method	**FPR**	**DR**	**ACC**	**E.R**	**FPR**	**DR**	**ACC**	**E.R**
ID3 p = 0.05	0.77	0.98	0.97	0.03	0.73	**0.99**	0.97	0.03
ID3 p = 0.07	**0.47**	0.99	**0.98**	0.02	**0.42**	**0.99**	**0.99**	0.01
CHAID p = 0.05, 0.01	0.80	**0.99**	0.97	0.03	0.73	0.99	0.97	0.03
CHAID p = 0.03, 0.02	0.80	0.98	0.97	0.03	--	--	--	--
C4.5 25, 2	0.70	0.98	0.97	0.03	0.42	**0.99**	**0.99**	0.01
C4.5 15, 2	0.80	**0.99**	0.97	0.03	--	--	--	--
C4.5 75, 2	0.61	**0.99**	0.97	0.03	0.71	**0.99**	**0.99**	0.01
I C4.5 25, 2	0.80	**0.99**	0.97	0.03	0.69	0.98	0.97	0.03
I C4.5 15, 2	0.61	**0.99**	**0.98**	0.02	--	--	--	--
Rule C4.5 25	0.65	**0.99**	**0.98**	0.02	0.72	0.99	0.97	0.03
Rule C4.5 15	0.73	0.99	0.97	0.03	--	--	--	--
NN (MLP) 100, 5	0.74	0.99	0.97	0.03	0.85	0.98	0.97	0.03
NN (MLP) 200, 5	0.84	0.99	0. 97	0.03	--	--	--	--
NB Prior=0.5 UCCI	0.84	0.99	**0.98**	0.02	0.85	0.98	0.97	0.03
NB Prior=0.5 All C	0.80	0.98	0.97	0.03	0.84	0.99	0.97	0.03
RBDT 25, 2	**0.38**	0.98	**0.98**	0.02	**0.31**	0.96	0.97	0.03

5.1 Discussions

From above Fig.6 and Fig.7, the proposed learning algorithm is shown some good results compare to ID3, CHIAD, C4.5, NB. So the conjunction rules of both C4.5 and Naïve Bayes always show affective performance. The rule C4.5 is good for dataset-1 when confidence level=25, 0.94 accuracy is given and also FPR is low, but while testing 72% of data instances its performance is not well compare to total instances. In above all classifiers used in dataset-1, the FPR is very less by NN (MLP) only, but same NN will gill give poor results in detecting true positives (DR). The DR is recorded less for RBDT in both datasets, 0.98 (average) for dataset-1 with all kind of combinations and testing 50%, 72% and 100% samples(observations). The NB also good in accuracy and DR noted 0.95 after testing

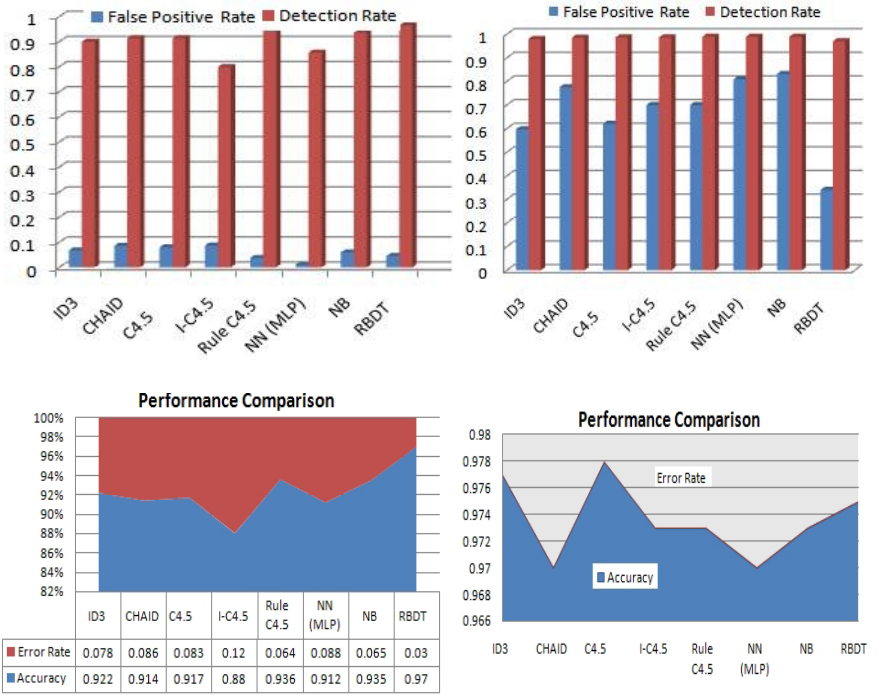

Fig. 6 The performance parameters and comparison for both datasets (1- Left, 2-Right) and classification algorithm on X-axis and rate of performance on Y-axis to be taken: The Fig (top left) shows the FPR and DR for dataset-1 and top right for dataset-2. Bottom left shows the performance comparison of all classifiers used in this paper (minimum 82%) for dataset-1and bottom right shows for dataset-2.

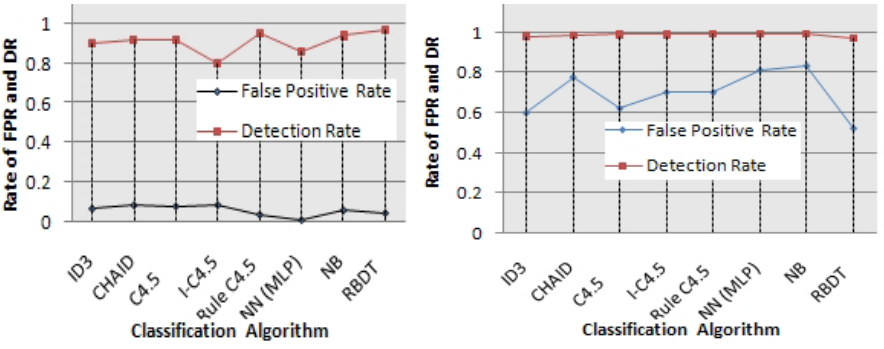

Fig. 7 The ROC is plotted for both datasets-1(left) and dataset-2(right) between FPR and DR with respect to classifiers

100% samples. For dataset-2, all algorithms not showing well performance in FPR, but C4.5, I-C4.5 and Rule C4.5 has given well in accuracy (0.99) and DR (0.98) for all kind of combinations and testing. The RBDT is also good for FPR measure, it noted only 0.35(average) for all test cases in dataset-2 and DR, accuracy for dataset-2 is almost got same results, even ID3 also. The error data in dataset-2 is problem, hence these kind results not repeated in dataset-1. Finally, our observations and combinations will help to improve the accuracy and detection rate with low false positive rate.

6 Conclusion and Future Work

The main objective (purpose) of this paper is to show some improve over the results by designing new rule based classifier. The RBDT classifier is novel rule based classifier based on partial conjunction and cascading rules from C4.5 and NB. In this paper we have constructed partial RBDT for dataset-2 and rules were simulated on both datasets, but in practical it is long and complex one. The RBDT is good for real time critical infrastructure applications. In this paper, the algorithm is only pointed conventional two child split. So in future work, we will extend it for more than two at each node. Finally our decision tree is modeled after strong investigation of research work in machine learning and the subject will focus in anomaly detection (AD) is fruitful to encourage everyone.

References

1. Denning, D.E.: An intrusion detection model. IEEE Transactions on Software Engineering (1987)
2. Axelsson, S.: Intrusion Detection Systems: A Survey and Taxonomy, Chalmers University. Technical Report 99-15 (March 2000)
3. Feng, H.H., Kolesnikov, O.M., Fogla, P., Lee, W., Gong, W.: Anomaly Detection Using Call Stack Information. In: IEEE Symposium on Security and Privacy 2003, CA, Issue Date: May 11-14, pp. 62–75 (2003) ISSN: 1081-6011 Print ISBN
4. Lee, W., Stolfo, S.J.: Data mining approaches for intrusion detection. In: 7th USENIX Security Symposium, Berkeley, CA, USA, pp. 79–94 (1998)
5. Lane, T., Brodley, C.E.: An Application of Machine Learning to Anomaly Detection. In: Proceedings of the 20th National Information Systems Security Conference, pp. 366–377 (October 1997)
6. Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001)
7. Jidiga, G.R., Sammulal, P.: Foundations of Intrusion Detection Systems: Focus on Role of Anomaly Detection using Machine Learning. In: ICACM - 2013 Elsevier 2nd International Conference (August 2013) ISBN No: 9789351071495
8. Jidiga, G.R., Sammulal, P.: The Need of Awareness in Cyber Security with a Case Study. In: Proceedings of the 4th IEEE Conference (ICCCNT), Thiruchengode, TN, India, July 4-6 (2013)
9. Kass, G.V.: An Exploratory Technique for Investigating Large Quantities of Categorical Data. Applied Statistics 29(2), 119–127 (1980)

10. Quinlan, J.R.: Induction of decision trees, Machine Learning 1, pp. 81–106. Kluwer Publishers (1986)
11. Quinlan, J.R.: Simplifying decision trees. International Journal of Man Machine Studies 27, 221–234 (1987)
12. Quinlan, J.R.: Decision Trees and Multivalued Attributes. In: Richards, J. (ed.) Machine Intelligence, vol. 11, pp. 305–318. Oxford Univ. Press, Oxford (1988)
13. Quinlan, J.R.: Unknown attribute values in induction. In: Proceedings of the Sixth International Machine Learning Workshop Cornell. Morgan Kaufmann, New York (1989)
14. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, Los Altos (1993)
15. Quinlan, J.R., Rivest, R.L.: Inferring Decision Trees Using The Minimum Description Length Principle. Information and Computation 80, 227–248 (1989)
16. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth Int. Group (1984)
17. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Prentice-Hall (2009)
18. Polat, K., Güne, S.: A novel hybrid intelligent method based on C4. 5 decision tree classifier and one against all approach for multi-class classification problems. Expert Systems with Applications 36, 1587–1592 (2009)
19. Jiang, S., Yu, W.: A Combination Classification Algorithm Based on Outlier Detection and C4. 5. Springer Publications (2009)
20. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the Twelfth International Conference on Machine Learning Chambery, France, pp. 115–123 (1993)
21. Yu, M., Ai, T.H.: Study of RS data classification based on rough sets and C4. 5 algorithms. In: Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series (2009)
22. Yang, X.Y.: Decision tree induction with constrained number of leaf node. Master's Thesis, National Central University (NCU-T), Taiwan (2009)
23. Michael, J.A., Gordon, S.L.: Data mining technique for marketing, sales and customer support. Wiley, New York (1997)
24. Loh, W.Y., Shih, Y.S.: Split selection methods for classification trees. Statistica Sinica 7, 815–840 (1997)
25. Clark, P., Niblett, T.: The CN2 induction algorithm. Machine Learning 3, 261–283 (1989)
26. Clark, P., Boswell, R.: Rule induction with CN2: Some recent improvements. In: Kodratoff, Y. (ed.) EWSL 1991. LNCS, vol. 482, Springer, Heidelberg (1991)
27. Rakotomalala, R., Lallich, S.: Handling noise with generalized entropy of type beta in induction graphs algorithm. In: Proceedings of International Conference on Computer Science and Informatics, pp. 25–27 (1998)
28. Chauchat, J.H., Rakotomalala, R., Carloz, M., Pelletier, C.: Targeting customer groups using gain and cost matrix: a marketing application. In: Proceedings of Data Mining for Marketing Applications Workshop (PKDD), pp. 1–13 (2001)
29. Rakotomalala, R., Lallich, S., Di Palma, S.: Studying the behavior of generalized entropy in induction trees using a m-of-n concept. In: Żytkow, J.M., Rauch, J. (eds.) PKDD 1999. LNCS (LNAI), vol. 1704, pp. 510–517. Springer, Heidelberg (1999)
30. Rajeswari, P., Kannan, A.: An active rule approach for network intrusion detection with enhanced C4.5 Algorithm. International Journal of Communications Network and System Sciences, 285–385 (2008)

31. Ghosh, A., Schwartzbard, A.: A study using NN for anomaly detection and misuse detection. Reliable Software Technologies, http://www.docshow.net/ids/useni
32. http://www.mathworks.in/products/matlab/
33. http://eric.univ-lyon2.fr/~ricco/sipina.html
34. Benferhat, A.S., Elouedi, Z.: Naive Bayes vs Decision Trees in Intrusion Detection Systems. In: Proc. ACM Symp. Applied Computing (SAC 2004), pp. 420–424 (2004)
35. Rokach, L., Maimon, O.: Decision Trees
36. Bache, K., Lichman, M.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, CA (2013), http://archive.ics.uci.edu/ml
37. Usta, I., Kantar, Y.M.: Mean-Variance-Skewness-Entropy Measures: A Multi-Objective Approach for Portfolio Selection. Entropy 13, 117–133 (2011), doi:10.3390/e13010117
38. Abdelhalim, A., Traore, I.: Converting Declarative Rules into Decision Trees. In: Proceedings of the World Congress on Engineering and Computer Science, Vol-I WCECS 2009, San Francisco, USA, October 20-22 (2009)
39. Abdelhalim, A.: Issa Traore, The RBDT-1 method for rule-based decision tree generation. Technical report (ECE-09-1), University of Victoria, STN CSC, Victoria, BC, Canada (July 2009)
40. Siva, S., Sindhu, S., Geetha, S., Kannan, A.: Decision tree based light weight intrusion detection using a wrapper approach. Elsevier-Expert Systems with Applications 39, 129–141 (2011), doi:10.1016/j.eswa.2011.06.013
41. Lowd, D., Davis, J.: Improving Markov Network Structure Learning Using Decision Trees. Journal of Machine Learning Research 15, 501–532 (2014)
42. Zaidi, N.A., Cerquides, J., Carman, M.J.: Alleviating Naive Bayes Attribute Independence Assumption by Attribute Weighting. Journal of Machine Learning Research 14 (2013)
43. Anchiang, D., Chen, W., Fanwang, Y., Jinnhwang, A.: Rules Generation from the Decision Tree. Journal of Information Science and Engineering 17, 325–339 (2001)