MINIG RULES OF CONCEPT DRIFT USING GENETIC ALGORITHM

P. Vivekanandan[1] and R. Nedunchezhian[2]

[1]*Dept. of Comp. Sci. and Engg.*
Park College of Engineering and Technology
Coimbatore, India
e-mail: anandpvivek@yahoo.co.in

[2]*Dept of Comp. Sci. and Engg.*
KalaignarKarunanidhi Institute of Technology
Coimbatore, India
e-mail:rajuchezhian@gmail.com

Abstract

In a database the data concepts changes over time and this phenomenon is called as concept drift. Rules of concept drift describe how the concept changes and sometimes they are interesting and mining those rules becomes more important. CDR tree algorithm is currently used to identify the rules of concept drift. Building a CDR tree becomes a complex process when the domain values of the attributes get increased. Genetic Algorithms are traditionally used for data mining tasks. In this paper, a Genetic Algorithm based approach is proposed for mining the rules of concept drift, which makes the mining task simpler and accurate when compared with the CDR-tree algorithm.

1 Introduction

Today's data sets like stock market, customer buying are liable to concept drift. Concept may drift due to the changes in the under lying context [1] and may happen because of several reasons. Concept drift can cause serious deterioration in the performance of the classification model built based on the old concept. Current research concentrates on correcting the models up-to-date [2, 3, 4, 5, and 6] based on the new data. But some times the rules of concept drift describing how the concept drifts may be interesting and useful. For example in a super market, customer buying pattern may change due to his, health condition or job loss. A fall of stock prize can be attributed to fraud in the company or recession. A sudden change in web access pattern can put a person in a terror suspect list. Very few researches have been focused on proposing a suitable method for mining the rules of concept drift [7]. Genetic Algorithm (GA) is a stochastic search method introduced in "Adaptation in natural and artificial systems", book written by Johan Holland. Nowadays it is widely employed as base algorithm for classification techniques in the fields of machine learning and data mining. The two initial and vital GA based classification algorithms are Learning Classifier Systems (LCS)[11 and 21] and Genetics-Based Machine Learning (GBML)[22]. Inspired by the above two methods, data mining community has developed several classification methods which employed GA as the core algorithm [8, 9, 10, 12 and 13]. The rules that GA find are more general because of its global nature. GA performance is better in complex domains because its model building process is based only on the natural principles and not on the domain knowledge. It operates in an iterative improvement fashion where search starts in a random location and gradually moves towards the regions of the model representation space that will produce good classification behavior. GA can successfully adapt to concept drifts [23] that occur in data streams in a fast and more natural way without any detection mechanisms. Hence, all the above mentioned properties of GA proves that it is a suitable natural tool for mining complex rules of concept drift in any data stream.

In the literature very little work employs GA to mine data streams with concept drift and no work in the literature uses GA to mine rules of concept drift. In this paper a classification method is proposed which does not tend to detect concept drift and adjust the classification model based on the new concept. Instead, it tries to identify and mine how the concept drifts, by using GA and represent the findings as a set of rules which can be called as "rules of concept drift". The paper is organized as follows section 1 defines the rules of concept drift with an example, section 2 describes previous work, and section 3 describes the proposed methods. Section 4 describes the experimental results and section 5 concludes the paper.

2 Rules of Concept Drift

Consider the diagnostic data sets (Table 1 and Table 2) for five patients observed at different time interval. Each record has five attributes namely patient ID, Exercise ("Yes" and "No"), Diet ("Yes" and "No"), Body-Mass ("Correct", "Medium", "Overweight" and "Obesity") and Sugar-Level ("High" and "Low"). When Sugar-Level is high the patient is in risk of diabetics.

Table 1. Patient Diagnostic Data

ID	Exercise	Diet	Body – Mass	Sugar– Level
1.	No	No	Correct	Low
2.	Yes	Yes	Correct	Low
3.	Yes	Yes	Correct	Low
4.	Yes	Yes	Correct	Low
5.	Yes	Yes	Correct	Low

Table 2. Patient Diagnostic Data Set after Some Interval

ID	Exercise	Diet	Body – Weight	Sugar– Low
1.	Yes	Yes	Over Weight	Low
2.	Yes	Yes	Correct	Low
3.	No	No	Over Weight	High
4.	Yes	Yes	Over Weight	Low
5.	No	No	Obesity	High

Consider the patients ID3 and ID5. By comparing them, it can be found that when Exercise change from "Yes" to "No" , Diet from "Yes" to "No" and the Body-Mass index from "Overweight" to "Obesity", the Sugar-Level changes from "Low" to "High". So the risk of diabetes is high. This is a simple example for rules of concept drift. Change in the attributes diet, Body mass and Exercise causes a drift in the concept. Here it was easy to identify the rules but in large data sets it will be a difficult task. The rules of concept drift can be written in the form If (Exercise = "Yes" "No") and (Diet = "Yes" "No") and (Body mass = "Overweight" "Obesity") Then (Sugar- Level="Low" "High")

3 Previous Work

There are several efficient algorithms available in the data mining literature that tends to correct the classification model when ever a concept drift is detected in the data set [2, 3, 4, 5, and 6]. But, so far only one algorithm has be proposed that tend to mine the rules of concept drift and was proposed by hen-I Lee et al [7]. They call their concept drift rule mining method as CDR tree algorithm. Their CDR tree algorithm first integrates new and old instance of the data from different time point into pairs. Then it creates a decision tree like structure called CDR tree and the tree building process is based on the principles of decision tree algorithm C4.5. The performance of the algorithm is quit well when the number of attributes and their corresponding domain values are small. But, its performance and accuracy degrades when the problem domain contains large number of attributes because of its complex tree building procedure.

To understand the complexity, Consider the example data set proposed in table 1 and table 2, which has three attributes{ Exercise(2 values), Diet(2 value) Body mass(4 values) and a target attribute sugar level(2 values). According to CDR tree definition each path of the tree will have a node for all the attributes of the problem domain. Each branch of a node represents a change in the attribute value or no change in the value. The leaf node represents whether there is a change in the concept value or not. So, if a test attribute of a node in the proposed tree has n values then the particular node will have n*n branches. For the example problem, let Exercise be the chosen test attribute for the root node of the CDR tree. Since the number of attribute values in the domain of the attribute Exercise is two ("No", "Yes"), the root will have four branches as described below:

$$\begin{array}{ccc} \text{No} & \rightarrow & \text{Yes} \\ \text{No} & \rightarrow & \text{Yes} \\ \text{Yes} & \rightarrow & \text{No} \\ \text{No} & \rightarrow & \text{No} \\ \text{Yes} & \rightarrow & \text{Yes} \end{array}$$

Suppose if there are ten attributes in a problem and if each attribute has ten values then the root node will have maximum 10*10 branches. A path in the tree will have maximum ten nodes and each node will branch 10*10 branches maximum. So some times the CDR tree becomes too complex to construct. Pruning methods must be employed to get a compact tree structure which may have impact on the accuracy of the model.

To reduce the complexity the best way for mining rules of concept drift is by using GA because GA' s rule discovery process is independent of the domain knowledge and is based only on natural evolution. The proposed method in this paper uses GA as base algorithm for mining rules of concept drift.

4 Proposed Method

In Genetic Algorithm based classification methods, the potential solution is represented by individuals called candidate rules [8, 9, 10, 11, 12 and 13] and are of the form

$$A1 \wedge A2 \ldots An \text{ THEN C}$$

The antecedent part of the rule is the conjunction of conditions say A (conjunction of attribute value pairs A1, A2... An) and the consequent part C is the class label. Each conjunction represents, how an attribute value changes due to concept drift and the consequent represents how the target attribute is affected by concept drift.

The classification process starts from a population of candidate rules which are called as chromosomes. Initial population is generated in a random manner and is processed in sequential steps, until an accurate solution gradually emerges. The sequential steps are called as generations. Each generation goes through the selection (or testing) phase and the reproduction phase. During the selection phase, candidate rules accuracy (fitness) is evaluated using the training data set and during the reproduction phase new chromosomes are created by applying genetic operators to the best chromosomes selected from the population. Following sub sections describes each of the process involved in the proposed GA based method.

4.1 Encoding

A population is created initially in a random manner in which each chromosome represents a candidate rule of concept drift. Let there be n input attributes and one target attribute. All the chromosomes of the population are of same size and will contain n genes. Each gene corresponds to an input attribute and is partitioned into three fields namely flag F_i, initial value IV_i and change value CV_i as show in Table 3. The flag field indicates whether or not the corresponding attribute value will be included in the rule or not. A value one shows that the attribute value will be included in the rule antecedent and a value zero shows that the attribute value will not be included. IV and CV fields indicate the change in the attribute value due to concept drift, when the records of two blocks of the training data that are generated at different time intervals are compared. Suppose, if an attribute has m values, then the IV and CV part of the gene representing the attribute in the chromosome will contain m bits each. Each value in the domain of the attribute will have a bit associated to it in the both IV and CV fields. IV field represents the value of the attribute with respect to records in the first block and CV field represents the value of the same attribute with respect to the records in the second block of data produced after concept drift. Among m bits only one bit will be set in both IV and CV part of a gene indicating a change of that attribute value from one value(Indicated by the IV field) to the other (Indicated by the CV field) due to concept drift.

Consider a set of a training data set; with three attributes A1, A2, A3 and a target attribute A4. Let them take 3, 4, 2 and 3 values respectively. Table 4 represents the chromosome coding and its meaning. Since Flag bit of A2 is Zero, A2 does not take part in the rule.

The consequent C_{xy} of a rule represents a change in value from x to y of target attribute. It is not encoded in the chromosome. It will be determined by the proportional situation of the training examples that the rule matches. The records in the training data set which matches a rule are noted and the class of those records is analyzed to find whether there is any class change. Usually multiple class change will match a rule. The class change that has the maximum number of match is assigned as consequent for that rule.

4.2 Crossover

Multi point crossover method is employed to generate the child population. Two chromosomes are selected randomly from the population, for reproduction, using roulette wheel selection method [14]. The genes of the selected chromosomes are exchanged with the probability called crossover probability (P_c=0.8).

4.3 Mutation

Mutation is applied for both CV and IV fields of all the genes for all the chromosomes with a probability called mutation probability (P_m). There are some restrictions for mutation. If a bit is selected from the CV or IV field of a gene, it will be set to one and all other bits of the corresponding CV or IV field will be set to zero. This ensures that in both IV and CV fields, only one bit will be set and it represents a change from one value to the other .The probability of mutation is governed by the mutation probability [Equation 1]. It is dynamic. It depends upon the fitness value of the chromosome according to equation1. Initially it is .5. It ranges between 0.0 and .5.

Mutation-rate =

$$= (1 - FitnessoftheSelectedchromosome)/2 \quad (1)$$

4.4 Selection

Best n elite chromosomes are chosen from the parent chromosomes for the next generation population pool. Remaining chromosomes for the new population are chosen from the new child population which has been created by the genetic process. The GA process is continued and stopped such that there is no significant increase in the fitness function for ten generations. It can also be stopped after specific number of generations has been elapsed. The final population is produced as output. Overlapping rules are removed manually and the final rule set is generated.

4.5 Fitness Function Calculations

A simple fitness function calculation called confidence factor [14] is used in the proposed method. Confidence factor (CF) can be defined as:

$$CF = |AandC| \,/\, |A| \quad (2)$$

Where A is the number of examples satisfying all the conditions in the antecedent part(input attributes) and A&C is the number of variables satisfying the antecedent A and consequent C(Target attribute)

4.6 Insert and Remove Operator

Insert and Remove operators control the size of the rule. Insert operator activates the gene by setting its flag bits and Remove operator deactivates a gene by resetting the flag bits with a varying probability (Pi and Pr). The probabilities range from 0 to 0.3 based on the number of genes that take part in the rule.

4.7 Proposed Algorithm

The steps in the proposed algorithm is explained below

Table 3. Chromosome Encoding

Attribute 1	Attribute 2	Attribute i	...	Attribute n
$F_1IV_1CV_1$	$F_2IV_2\ CV_2$	$F_i\ IV_i\ CV_i$	...	$F_n\ IV_n\ CV_n$

Table 4. Example Chromosome and its equivalent rule

Attribute	A1	A2	A3	
Code	1001100	000010010	11001	
IF A1= Value3	->	Value1 and A3= Value1	->	Value2

1. Start – Generate random population of n chromosomes (Section 3.1)
2. Fitness – Evaluate the fitness of each chromosome in the population (section 3.5).
3. New population – Create a new population by repeating following steps until the new population is complete
    1. Selection – Select two parent chromosomes from a population according to their fitness(section3.2).
    2. Crossover – With a crossover probability cross over the parents to form new offspring (section 3.2).
    3. Mutation – With a mutation probability mutate new offspring (Section 3.3).
    4. Insert and Remove Gene – Apply Insert and remove operators (Section 3.6)
5. Accepting – Select new population for a further run of the algorithm (Section 3.4).
6. Test – If the end condition (section 3.4) is satisfied, stop, and return the current population as best solution, otherwise go to step 2.

5 Experimental Study

5.1 Stagger Data set.

A modified form of Stagger data set [16, 17 and 18] is used to simulate concept drift. Each record in the data set has three attributes: color {“green”, “blue” and “red”}, shape {“triangle”, “circle”, and “rectangle”} and size {“small”, “medium” and “large”}. There are four concept classes namely: - class A: if color = “red” and size=”small”; class B: if color =”green” and size =”medium”; and class C: if color=”blue” and size=”large” and other cases it is class D.

Thousand records where randomly generated for a block. The data set is built such that 20% of the records belongs to class A and 20% belongs to class C. Two such blocks are created. The first block is considered as initial block. In the second block 80% of class A records are selected randomly and are converted to class B records. Similarly class C records are also converted to class B records in the same manner. Initial population is created with 30 chromosomes and 5 elite chromosomes are copied to next generation.

Table 5. Important rules generated

Rule	Fitness
IF $(Color = blue \rightarrow blue) and (Shape = triangle \rightarrow triangle) then (classD \rightarrow classD)$	1.0
IF $(Color = red \rightarrow green) and (Size = small \rightarrow medium) Then (classA \rightarrow classB)$	.8
IF $(Color = blue \rightarrow green) and (Size = large \rightarrow medium) Then (classC \rightarrow classB)$	.8
IF $(Color = green \rightarrow green) and (Size = medium \rightarrow medium) Then (classB \rightarrow classB)$	1.0

The experiment is repeated until there is no considerable increase in the fitness of the chromosomes continuously for ten generations. From the final rule set overlapping rules are detected manually and are removed. In an average the rules are generated between generations 60 and 80. Table 5 describes some of interesting rules that are mined with their fitness function. The generations at which the rules are generated depend on the initial population. The results in the table (table 5) indicate that GA can mine concept drifting rules in a precise and accurate manner.

5.2 IBM Data Generator Data Set

IBM Data Generator [19] is a public and widely used data generator and it has several well-defined classification functions and parameters which can be used to generate different characteristics of datasets. The dataset generated by IBM Data Generator contains one Boolean target class and nine basic attributes: salary, commission, loan, age, zipcode, hyears, hvalue, elevel, and car. Among the nine attributes, zipcode, elevel, and car are categorical attributes, and others are the continuous attributes. In our experiment, four classification functions, P3, P5, P43, and P45 are randomly selected to generate the experimental datasets. In order to analyze the performance of CDR-Tree and CityplaceProposed StateGA based method under different drifting ratio $R\%$ (i.e. the proportion of drifting instances to all instances), we use the four functions mentioned above to generate required experimental datasets. For each function, the noise level is set to 5% and the dataset generated by IBM Data Generator is regard as the original/first dataset in the data stream. Then using the first dataset a second data set is generated by introducing drift to its records. To introduce a drift, $R\%$ (R = 5, 10, 15, 20, 30) values of each attribute, including the target attribute are replaced by a random value in the domain. As a result, each function will generate five new datasets with different drifting ratio. A total 4 old datasets and 20 new datasets are generated in our experiments. Every dataset includes 10000 instances and the 10-fold cross-validation test method is applied to all datasets. That is, each original dataset is divided into 10 parts of which nine parts are used as training sets and the remaining one as the testing set. Both the proposed GA and CDR-tree algorithm are used to mine the rules of concept drift. Both the results are compared and are described in the tables 6, 7, 8 and 9.

5.2.1 The Analysis

The second experiment shows that GA can be a successful tool for mining the rules of concept drift, particularly when the drift in the data is too complex. From the tables (tables 6, 7, 8 and 9) it can concluded that, rules generated by both the CDR-Tree algorithm and GA have a high accuracy in all the 20 datasets. But, if the concept drift level increases, the accuracy of the CDR-Tree drops.

Table 6. Accuracy of CDR tree and GA based rules for drift D(3)

Data set	Concept drift ratio %	Accuracy	
		CDR tree	GA
D(3)	5	92	92.5
	10	90.8	92.1
	15	89.2	92.3
	20	88.5	92.1
	30	86.2	92

Table 7. Accuracy of CDR tree and GA based rules for drift D(3)

Data set	Concept drift ratio %	Accuracy	
		CDR tree	GA
D(3)	5	90	91.5
	10	89	90.1
	15	87.2	91.3
	20	86.5	91.1
	30	85.2	90.2

Table 8. Accuracy of CDR tree and GA based rules for drift D(3)

Data set	Concept drift ratio %	Accuracy	
D (3)	5	90.1	90.5
	10	89.1	90.1
	15	88.2	90.2
	20	86.5	90
	30	84.2	90

A higher concept drift level would make the CDR-Tree building process more complex and this caused the drop in the accuracy. From the tables it can be seen that the accuracy of the rules produced by GA does not drop even when the drift level increases. This is due to the fact that the problem domain complexity does not effect the GA process because, GA process is entirely based on natural selection and reproduction and completely independent of problem domain.

Table 9. Accuracy of CDR tree and GA based rules for drift D(3)

Data set	Concept drift ratio %	Accuracy	
D(3)	5	87	88
	10	86	87.1
	15	85.1	87.2
	20	84.5	87
	30	83.2	87.2

5.3 Experiment 3

The third experiment is performed using 23 public benchmark datasets taken from the UCI Machine Learning Repository and the UCI KDD archive and are described in table 10.

The continuous attributes are converted to categorical attribute using binning strategy with equi-width 5 bins for experimental purpose. The original data set forms the initial block of data. To generate the second block, 50% of the attributes of every data set is chosen randomly to induce concept drift which also includes the target attribute. Then, 30 % of the records from every data set is selected randomly and the values of the chosen attributes of the particular data set is replaced with an another random value selected from the corresponding attribute domain. The new block so obtained is joined with the first block to form the test data set. The proposed GA based method is applied to mine the rules of concept drift as described in section 3. To check the efficiency of the proposed method 10-fold cross-validation test method is applied as described in section 4.2. For all the datasets CDR-tree is generated for comparison. The results are tabulated in table 10.

5.3.1 The Analysis

From the table 11 it can be proved that the proposed GA based methods accuracy is better when compared with the CDR tree algorithm. For data sets with less number of attributes the accuracy is almost same for both the data sets. As the number of attribute increases the accuracy of the CDR tree falls considerably when compared to the proposed GA based method.

Table 10. Experimental data set description

Index	Dataset name	# of instances	# of attributes	# of classes
1	Network intrusion	49270	41	5
2	USPS Data	9298	256	10
3	Nursery	12960	8	5
4	Solar Flare	1389	10	6
5	Yeast Database	1484	8	10
6	Car	1728	6	4
7	Image segmentation	2310	19	7
8	Thyroid	3772	28	4
9	Page blocks	5473	10	5
10	Optical Digits	5620	64	10
11	Satimage	6435	36	6
12	Isolet Spoken Letter	7797	617	26
13	LED Display	10000	7	10
14	Waveform	10000	21	3
15	Pen Digits	10992	16	10
16	Australian sign language	12546	8	3
17	Letter	20000	16	26
18	Poker	25000	10	10
19	Chess (King RootKing)	28056	6	18
20	Shuttle	58000	9	7
21	Connect-4	67557	42	3
22	Adult	48842	14	2
23	German credit data	1000	20	2

Table 11. Error rate of CDR tree and GA. Average Classification Error

Index	Dataset name	CDR tree	IGA
1	Network intrusion	30.2.	18.1
2	USPS Data	6.4	6.2
3	Nursery	9.6	10.4
4	Solar Flare	15.2	15.8
5	Yeast Database	12.0	12.1
6	Car	5.4	6.2
7	Image segmentation	22.7	21.8
8	Thyroid	5.0	3.4
9	Page blocks	12.0	10.5
10	Optical Digits	19.6	11.8
11	Satimage	20.6	10.8
12	Isolet Spoken Letter	22.9	15.2
13	LED Display	11.1	12.3
14	Waveform	16.3	8.0
15	Pen Digits	15.6	11.5
16	Australian sign language	11.1	11.1
17	Letter	14.7	11.0
18	Poker	14.5	12.1
19	Chess	12.2	11.6
20	Shuttle	14.2	12.8
21	Connect-4	27.9	25.8
22	Ipums-la-99	16.4	17.9
23	Forest Cover type	24.2	18.4

6 Conclusion

An important task in the classification process of data with concept drift is to update the classification model when ever a concept drift is detected to prevent the drop in the accuracy level. Recent research mainly focuses on it. But rules of concept drifts are also interesting and useful some time. In this paper a method has been proposed to mine the rules of concept drift using Genetic Algorithm. The experimental study shows the success of the proposed algorithm. The experiments also demonstrate the superiority of the proposed method by comparing it with the existing CDR tree algorithm. In future the accuracy of the proposed algorithm will be demonstrated with a real data set and will be compared with the CDR tree. There is also a need to propose a proper initialization method for the proposed algorithm.

References

[1] G. Widmer and M. Kubat, *Learning in the presence of concept drift and hidden contexts*, Machine Learning, 23(1):69–101, 1996.

[2] A. Tsymbal, *The problem of concept drift: definitions and related work*, Department of Computer Science, PlaceNameTrinity PlaceTypeCollege CityplaceDublin, Tech. Rep. TCD-CS-2004-15, 2004.

[3] H. Wang, W. Fan, P. S. Yu, and J. Han, *Mining concept-drifting data streams using ensemble classifiers*, in Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 226–235 , 2003.

[4] G. Hulten, L. Spencer, and P. Domingos , *Mining time-changing data streams*, Proceedings of 7th ACM SIGKDD International Conference on Knowledge and Data Mining, 97–106, San Francisco, CA, ACM Press, 2001.

[5] Mihai Lazarescu, Svetha Venkatesh and Hai Hung Bui *Using Multiple Windows to Track Concept Drift* , Intelligent Data Analysis Journal, 8(1), 2004.

[6] S. U. Guan and F. ZhuCollard, *An incremental approach to genetic-algorithms based classification. Systems*, Man and Cybernetics, Part B, IEEE Transactions, 35(2):227–239, 2005.

[7] Chen-I Lee, Cheng-Jung Tsai, Jhe-Hao Wu and Wei-Pang yang, *A Decision Tree-Based Approach to Mining the rules of Concept Drift* , Fourth International Conference on Fuzzy System and Knowledge discovery, 2007

[8] K. A. De Jong., W.M. Spears and D.F. Gordon, *Using genetic algorithms for concept learning*, Machine Learning 13:161–188, 1993.

[9] C. Z. Janikow, *A knowledge-intensive genetic algorithm for supervised learning*, Machine Learning 13:189–228, 1993.

[10] D. P. Greene and S. F. Smith, *Competition-based induction of decision models from examples*, Machine Learning 13:229–257, 1993.

[11] K. A. De Jong and W. M. Spears, *Using genetic algorithm for supervised concept learning* , Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence, 1990.

[12] E. Noda, A.A. Freitas and H.S. Lopes, ‘‘*Discovering interesting prediction rule with a genetic algorithm* , Proceedings of the 1999 Congress on Evolutionary Computation, 2, 1999

[13] I-hui li, I-en liao and Wei-zhi pang, *Mining classification rules in the presence of Concept drift with an incremental genetic Algorithm* , journal of theoretical and applied information technology, 2008.

[14] Alex A Freitas, *A Survey of Evolutionary Algorithm for Data Mining and Knoweldge Discovery*, Advances in evolutionary computing: theory and applications, 819–845, ACM press, 2003.

[15] Linyu Yang, D. H. Widyantoro, T. Loerger T, J. Yen, *An Entropy-based Adaptive Genetic Algorithm for Learning Classification Rules*, IEEE proceding of the 2001 congress on Evolutionary computing 2:790–796, 2001.

[16] Shixi Chen, Haixun Wang, Shuigeng Zhou, Yu P.S., *Stop chasing trends: discovering high order models in evolving data*, IEEE 24th International Conference on Data Engineering, 2008.

[17] Y. Yang, X. Wu and X. Zhu, *Combining proactive and reactive predictions for data streams*, in SIGKDD, 710–714, 2005.

[18] Jeremy Z. Kolter, Marcus A. Maloof, *Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift*, IEEE International conference on Data Mining, 2003.

[19] R. Agrawal, A. Ghosh, T. Imielinski, B. Iyer and A. Swami, *An interval classifier for database mining applications*, In proc. of the 18th Conf. on Very Large Databases, August 1992.

[20] J. H. Holland, *Adaptation in natural and artificial systems*, PlaceTypeUniversity of PlaceNameMichigan Press, CityplaceAnn Arbor, 1975.

[21] Wilson SW, *Classifier fitness based on accuracy.* Evolutionary Computation (3):149–17, 1995.

[22] Freitas A. A., *Data Mining and Knowledge Discovery with Evolutionary Algorithms.* Springer-Verlag, 2002.

[23] P. Vivekanandan R.Neduchezian, *Mining data streams with concept drifts using genetic algorithm*, Artificial Intelligence Review, Springer-Verlag, Accepted Available online, 2011.