Data Mining Classification: *The Potential of Genetic Programming*

Nabil H. El Kadhi
Computer Engineering Department Chairperson
AHLIA University - Manama, Bahrein
EPITECH- Kremlin Bicêtre FRANCE
nelkadhi@ahliauniversity.edu.bh

Fatima A. Habib
MITCS Program Student
AHLIA University, Manama, Bahrein
ARAMCO Saudi Arabia
fatima.habib@aramco.com

Abstract—**Data Mining (DM) is one of the techniques used for the process of searching through a huge volume of data in a database to uncover useful and interesting information. Classification is a commonly studied issue in data mining. It involves predicting the categorical attribute (class) value on the basis of other attributes (predicting attributes) values. One of the classification approaches to data mining is gene expression programming (GEP), which is a development of Genetic algorithm (GA) and Genetic Programming (GP). In this paper, we investigate the potential of genetic programming for data mining classification. It is therefore important to investigate the advantages and challenges associated with using tree-based Genetic Programming algorithms for data mining classification. This study determines how better data mining classification performance can be achieved using genetic programming. It demonstrates how the search scope can be refined through heuristics and machine learning methods to reduce and change the search space for our Genetic Programming classifiers. A specific design and application of GP to two classes data analysis is presented as well as a set of experimental results showing the efficiency of the suggested application.**

Keywords - Data Mining; Genetic Algorithm; Genetic Programming; Classification.

I. INTRODUCTION

In this paper, we are interested in analyzing a large volume of data. With huge data volume we have a huge potential for discovering knowledge through prediction and description. In the same time it creates a problem because of the difficulty of knowledge extraction and data classification. Therefore, there is a need to develop intelligent and powerful tools to derive interesting, specific, and useful information out large databases. Well, a number of techniques have been developed to aid in data mining [11] and have even proved to be practical. It is important, however, to note that data mining involves a number of tasks and not all these tasks are suitable for data analysis. In light of this, there has been growing interest on classification as a suitable approach to data mining leading to an emergence of a number of techniques that can be used in the classification task. Some researchers have continuously argued that techniques that use genetic algorithm (GA) are superior to conventional approaches in data mining classification in regard to performance, search space and time. Note that there has also been an emergence of several GA-based approaches and system for data mining classification. Genetic programming has evolved out of GA concepts, and has increasingly been accepted as a suitable classification technique. Most researchers have not explored extensively variant GP approaches to generally validate GP as a suitable method to data mining classification. This paper does not negate the usefulness of other approaches but investigates the potential of GP as a classification technique. Therefore, it uses the SQL-based GP to validate it as the most suitable technique for purposes of data mining classifications. GP is emerging as a powerful technique in deriving knowledge from large databases. The objectives of the study are:

- To show the potential of GP in induction of classifiers and validate the technique as suitable in data mining.
- To show that the search scope can be reduced through heuristics and GP techniques to facilitate data mining classification in large databases.
- To illustrate the process and implementation of data mining classification through a GP-based technique.
- To investigate the challenges and advantages of using the tree-based GP algorithms in data mining classification.

The paper is organized as following. Section II review some related works. Section III come through the proposed system design including all the Genetic algorithm parameters and organization. Section IV deals with implementation aspects and test results before concluding in Section V.

II. RELATED WORKS

Saraee and Sadjay [9] presented three approaches for classification via GP and illustrated that GP is superior to the traditional techniques for classification in terms of performance on space and time required for processing. Thus, with GP it is possible to create optimized classification rules in a manner that the traditional techniques cannot accomplish at low cost. Furthermore, the method allows the comparison of attributes of same type within a data set by generating specific set of rules. Ten et al. [10] have introduced a concept mapping method for evaluating the fitness of individuals through improved representation of booleanized attributes and token competition. The approach applies a covering algorithm that uses a memory vector similar to an artificial immune system to generate several rules while eliminating the redundant ones. This GP classifier is authenticated upon a number of benchmark datasets chosen from the UCI Machine Learning Repository. Thus, the GP approach with a basic tree structure of only 'AND' and 'NOT' functions is effective in representing solutions in classifications. Furthermore, this approach is capable of generating understandable rules from datasets even when the background information is lacking on the particular dataset. On the other hand, the approach indicated low performances on some datasets, which nonetheless, does not disqualify the GP classifier as a potential method in data mining.

A new variant of linear genetic programming where genetic programs are characterized by the linear sequences of a C programming language is introduced in [1]. This system uses an algorithm that eliminates instructions that are not effective (the introns) prior to the execution of a program during fitness processing. This allows a substantial acceleration in the implementation speed, which is significant with composite data sets since they are being used in real-time.

Folino et al. [5] introduced a variant of cellular genetic programming that uses the boosting and bagging techniques to induce a group of predictors in data classification. Bagging is based on bootstrap samples or replicates of same size of the training set. Boosting enhances the performance of the weak learning algorithm. Carreno et al. [3] conclude further that REC has obtained high performance in building understandable classifiers with impressive predictive quality. Muharram and Smith [7] showed that genetic programming is efficient in generating highly predictive non-linear features using the novel attribute group. Espejo et al. [4] presented a number of solutions related to the use of GP in classification purpose have been evaluated. It revealed that the flexibility of this method makes it suitable in formulation of classifiers as well as in some pre-processing and post processing tasks. However, these researchers identify high training time as a drawback of GP. Furthermore, this becomes worse when large amounts of data are involved. Sakprasat et al. [8] have demonstrated the usefulness of strongly-typed GP in automatic approval of credit. GP can be applied in data mining for a problem regarding automatic approval of credit, and it is effective even with data that lacks some values. GP has been shown to be a superior method of classification in terms of reduced amount of space and time required for processing. The approach has also been credited with reduced number of rules, which is achieved through comparison of same type attributes. GP has also the capacity to generate comprehensible rules even when some information is absent. So many GP techniques have been used in the market using classification.

III. Proposed System Design

This paper focus on using Genetic Programming for classification task in data mining. This section presents the design of the proposed GP model. It shows how various GP techniques would be incorporated into an algorithm that would be used in data mining classification. The various components of the GP algorithms are highlighted.

A. GP Model and Abstract Tree

The application of GP techniques, including the natural selection should allow generation of better data mining solutions. Therefore, this paper proposes a GP algorithm that has immense benefits in induction of classifiers. The process of induction will be characterized by random selection of a given population of computer programs that will be used to produce subsequent generation by applying genetic operators, as it is with biological evolution. A fitness function, which is a binary function, will be used to direct the process of the GP algorithm in selecting the computer programs or rather the individuals. A grammar-based abstract or derivation trees would be used in the representation of computer programs in this GP model using abstract trees [6]. The abstract trees have the advantage of restricting the destructive aspect of the genetic operators on computer programs such syntactically erroneous code is not created. A context-free grammar would be used as the basis for constructing the abstract trees. The trees would be characterized by a terminal and a non-terminal node. The terminal nodes comprise independent values, while the non-terminal nodes depend on its components evaluation. The generation of a suitable computer program from a specific grammar would require a random selection of non-terminals. Regarding the GP Program, an initial population will be randomly generated before running the GP algorithm in a continuous loop. The loop will be characterized by two activities:

- Evaluation of individual program by the specifically designed fitness function.
- Generation of a new population of computer programs through selection of these individuals depending on their fitness and the use of the genetic operators like reproduction, mutation and crossover. The general program model is presented Fig. 1.

The initial generation of programs uses the genetic algorithm (GA) shown Fig. 2. Each of the iterations represents a generation. Basically, the GA is generic but various components would be incorporated to allow it to perform specific tasks. These additional aspects of the GA include the representation of chromosomes, selection strategy, and genetic operators (crossover and mutation). Thus, arrays of bits are used to represent chromosomes (computer programs), while simulation evolution – the selection strategy – is used to process the individuals or rather the computer programs.

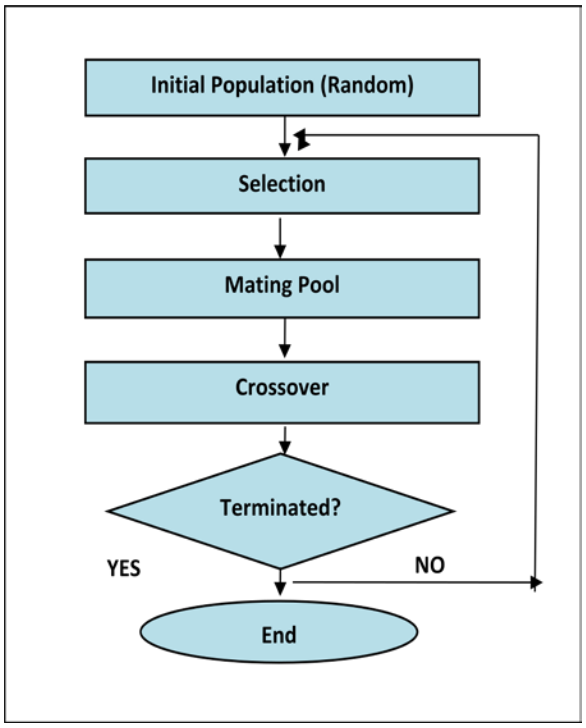


Figure 1. Our Genetic Program Model

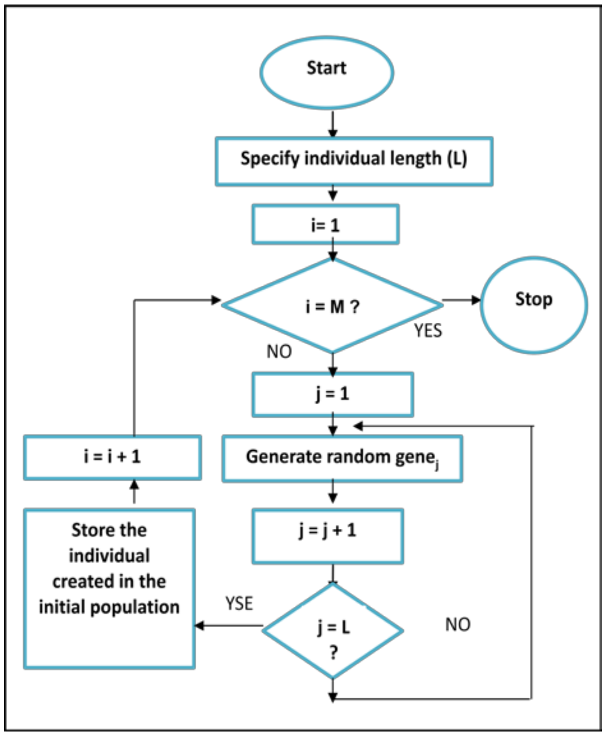


Figure 2. GP Selection Model (Algorithm)

B. *Selection Strategy*

Our GP model adopts a tournament selection approach so that various individuals are chosen randomly from a population for evaluation and the best individual in this group is designated as parents. This process is then repeated as often as there are individuals to choose. The selected parents are used in generation of new offsprings through uniform and random operation. The selection size will depend on the problem, population size, and so on. The parameter for selection is based on the population size. The tournament size ranges from two to a number equivalent to the number of individuals in the population. The selection would be done by selecting random indices - index1 and index2, where the model for both indices is characterized by classes 1 and 0 and having values that range from 1 to 100. This kind of selection is applied to minimize the selection pressure.

C. *Fitness Functions*

The fitness function is designed to direct the GP algorithm in selecting individuals (computer programs). The fitness functions do not change; they are non-mutable. A mutable fitness function will give different results and make comparison of the programs fitness difficult. The model demonstrates that through the use of various heuristic approaches with GP, the classification task in data mining can be simplified without compromising the integrity of the results. The systems is also useful in information retrieval (IR) applications such as document clustering; document matching and ranking, document indexing, query induction, optimization and representation. Fig. 3. shows the suggested fitness measurement function.

$$\text{Fitness} = (0.5{*}\text{acc }(0) + (.05{*}\text{acc }(1))$$

$$\text{Where accuracy for class } 0 = (f_{10} + f_{00}) / (f_{10} + f_{00} + f_{01} + f_{11})$$

$$\text{And the accuracy for class } 1 = (f_{01} + f_{11}) / (f_{10} + f_{00} + f_{01} + f_{11})$$

Figure 3. Used Fitness Function

The Fitness function is in fact a simple linear formula which designed specifically for the GP giving a result that will be either 0 or 1. Please notice that our reproduction selection act by copying the individual to next generation without making any changes to it. As shown Fig. 3, the different classes are measured by specific functions (f). The fitness function, is one of the most essential component in GP that measures how good an individual is as a solution to the problem. It could take several criteria into account and guide GP to seek the most preferable part of solution space by probably tuning the weight of the criteria involved. Before setting up the fitness function for GP, some measures involved have been introduced. this is because the study mainly focuses on binary classification. Fig. 3 functions show some of measures and illustrate the number of instances belonging to four possible categories after classifying each of examples in a given dataset f_{10} indicates the number of the False negative. Negative (summation of instances that were predicted 1 but they were actually 0). f_{00} indicates the number of the True Negative (summation of instances that were predicted 0 and they were actually 0). f_{01} indicates to the False Positive. (summation of instances that were predicted 0 but they were actually 1). f_{11} indicates the True positive. (summation of instances that were predicted 1 and they were actually 1). This is applied when a program operates efficiently, or when its fitness function is very high [9].

D. *Crossover*

The crossover involves one or two programs (parents) being used to produce two fresh offspring for the subsequent generation. The first offspring are selected through the tournament method, whereby segments of these offsprings are selected randomly and exchanged to give two offspring for the next generation. The crossover operator permits the exchange of segments that have been produced using the same rule of evolution. The crossover operation would be executed 30 times; this number represents the maximum number of generations l. A total of 50 crossovers would be performed between the two models until the best fitness is obtained. This implies that the solution performs crossover between indexes of the models 1, 2, 3, 4, 5, 6 until 50 child models are created. The crossover for class 0s would also be created from the two models in order to determine the new class 0 by calculating the number of rules in both sets of the models. If 4 or 3 rules are obtained then random numbers of rules are chosen and used in the predictive class. The rules should be between 1 and 3 if the sum of the rules is equal to 3. The results of the 3 rules are mixed together in order to choose the best rule to be applied/used in the predictive class. The rules were selected based on fraction.

The initial population is first considered as a whole. Covering algorithm are used here to ensure a high level of accuracy. The basic idea about covering algorithm is to maximize instances of the desired class and minimize the instances of other classes. In our model, we start with a general rule of: *IF X then Y=0*. The idea here is to replace X with an exact condition. After that, the rule having the maximum accuracy will be selected. Then we delete instances that do not satisfy the condition of the selected rule. The same technique will be repeated on the remaining data until we reach a rule with almost 100% of accuracy.

E. *Elitism*

Elitism involves copying the best chromosomes to the initial population. This is beneficial in preventing the loss of chromosomes with good quality during the genetic operation. Thus, it ensures that the best programs of present generation survive the subsequent generation depending on the rate of elitism. It helped in maintaining subpopulations close to local and global optima. Through elitism the best chromosomes would be maintained. We ensure elitism by storing the best fitness in the next generation until achieving a better fitness. The proposed GP model will culminate into developing a data mining tool that will use the techniques discussed above to induce classifiers. The tool or program incorporates genetic operators, SQL grammar and a fitness function.

F. *Analysis and Advantages*

The proposed system has a number of advantages over various existing data mining systems including those which use GP. To start with, the advantages of GP over other techniques of knowledge discovery in databases are well documented. It is important to note that there are a number of approaches to knowledge discovery in electronic databases. These include description, clustering, association, regression, and classification. Recent studies have shown that classification is a suitable approach to knowledge discovery in large databases [6]. The development of genetic programming has been hailed as a milestone towards creation of superior techniques that are able to mine useful information from databases ([2] and [9]). Thus, genetic programming can be used to solve a number of real world business problems.

The suitability of the genetic programming and classification as data mining techniques is a clear advantage of the proposed system over other systems that do not use GP in knowledge discovery. The proposed GP model has a number of advantages. The GP design, for instance, can handle both numerical and string data types. The use of SQL makes the tool suitable in mining databases since programs evolved are represented in the SQL language. The GP approach also gives high accuracy of results. Furthermore, the proposed tool can include optimization metric such as area under roc curve (AUC) and Brier score (BRI) [6]. Although the system is general purpose and may be able to handle specific problems. The use of the SQL to represent the computer programs has its advantages. The SQL is a declarative language, this makes it easier to use tree structures. Furthermore, in comparison to systems that use linear GP, the system is preferable. In Linear Genetic Programming, individuals are represented as sequence of instructions which makes execution of the instructions time consuming.

IV IMPLEMENTATION- EXPERIMENTAL RESULTS

The experimental prototype has been developed using VB.NET 2008 and MS Access. The records were stored in a single table in MS access. The dataset contained information collected by the US Census Service concerning housing in the area of Boston Massachusetts. It was obtained from the StatLib archive: (http://lib.stat.cmu.edu/datasets/boston). The dataset has 506 cases. There are 14 attributes in each case of the dataset. Table 1 shows the used attributes CATMEDDEV is an attribute which has been created by categorizing median value (MEDV) into two categories. The median housing price in housing tracts in the Boston area falls into the "high" or "low" category. Using '1' for high values of MEDV and 0 for low values. As already mentioned the tool was created in VB.NET 2008 to read the data, generate the initial population, perform mutation and determines the best fit values. Fig. 5. shows the system interface. For seek of Cross Validation the Dataset have been divided into two subsets in order to train data and to test data, therefore, cross validation technique is used to evaluate the GP results. In the project, dataset was divided to 70 % for training and 30% for testing and evaluation. After generating the initial population, the execution is done 50 times (cycles). Each execution was done twice, one for 0s models and for the 1s models. The initial population was created in the following order:

- Randomly choose the number of conditions.
- Randomly operate between each condition (<><=>=).
- Randomly obtain a number between the minimum and maximum of the attributes

The initial population model is between 0 and 1. Each gene in the initial population is generated as a random number, distributed uniformly over the range [0,n] (the productions are numbered from 0 to n, so that any production can be represented by a number in the range [0,n]). In this method, all individuals (genotypes) in the initial population have a fixed length. Therefore the process of generating each individual is done using the following steps:

1 Each gene is generated randomly.
2 Repeat step 1 until the number of genes of that individual reaches the pre specified length.
3 If all individuals in the initial population are created, then continue to perform other GP operations, otherwise, repeat steps 1-3.

One iteration of the algorithm is referred to as generation. The basic GA is generic and there are several aspects that can be implemented differently according to the problem. This can be achieved by representation of solution or chromosomes, adopting appropriate selection strategy, applying crossover and mutation operators.

TABLE 1. DATABASE ATTRIBUTES

Attribute	Description
CRIM	Crime rate
ZIN	Proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	Proportion of non-retail business acres per town.
CHAS	Charles River dummy variable (1 if tract bounds river; 0 otherwise)
NOX	Nitric oxides concentration (parts per 10 million)
RM	Average number of rooms per dwelling
AGE	Proportion of owner-occupied units built prior to 1940
DIS	Weighted distances to five Boston employment centres
RAD	Index of accessibility to radial highways
TAX	Full-value property-tax rate per $10,000
PTRATIO	Pupil-teacher ratio by town
B	1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
LSTAT	% lower status of the population
MEDV	Median value of owner-occupied homes in $1000
CATMED DEV	The predicted classes

GA is implemented by having arrays of bits that represent chromosomes. The individuals in the population are then processed by simulation evolution. Simple bit manipulation operators are then used to allow the implementation of crossover, mutation and other operations. For example, to generate the initial population for a random number 3 for 0 models. In the first round, this is chosen randomly between the operations of each condition. The result will be: Med<.07 and max<=8.9 and Dix<8.88 Where the numbers 0.07, 8.9 and 8.88 are between the minimum and maximum values of the attributes. The values for model 1 are generated in the same way. The results of the experiments involving the GP-SQL Mining Tool and the database containing the Boston housing information reveal the effectiveness of the system. To start with, the system was able to apply the genetic operators effectively to evolve programs and select the most suitable rule for classification. Thus, it was able to reproduce programs without any changes for the next generation, perform crossover mutation on programs. Furthermore, it allowed diversity in the population of programs to prevent an early derivation of algorithm solution. These genetic operators are also applied in LOGENPRO, Neural Net, and GPSQL Miner, which showed impressive results or accuracy in the experiments. In this regard, it is substantial, therefore, to comment that the GP is a suitable approach to classification or induction of classifiers in data mining.

The effectiveness of the system was tested against a large database of Boston housing data. The results showed that the system is promising in regards to its application in classification of data in the data mining process. It generated reliable results that are highly accurate when a large database that has a lot of noise was used. The system was further validated against other common GP-based approaches for data mining classification including GP-Knn and GPSQL Miner. The level of accuracy for the system is higher than that achieved by both the GPSQL Miner and GP-Knn. Furthermore, the literature revealed that the various methods integrated into the system are beneficial in maintain the best chromosomes, reducing the search space, and generating the most suitable program for classification. The system is advantageous in that it uses an interface that is user friendly, and at the same time maintains the advantages of using the SQL grammar in manipulating the database. However, the use of a high-level programming language increased the execution time needed compared to a system that is based on a low-level language. The accuracy of classifiers is shown in Table 2. Our system (labeled GP-SQL Mining Tool) shows higher accuracy than the other two systems; it is better than the least accurate of the three algorithms (LOGENPRO) by more than 5%. Thus, the GP-SQL Mining Tool gives better results, and therefore, is reliable and suitable for induction of classifiers in large databases.

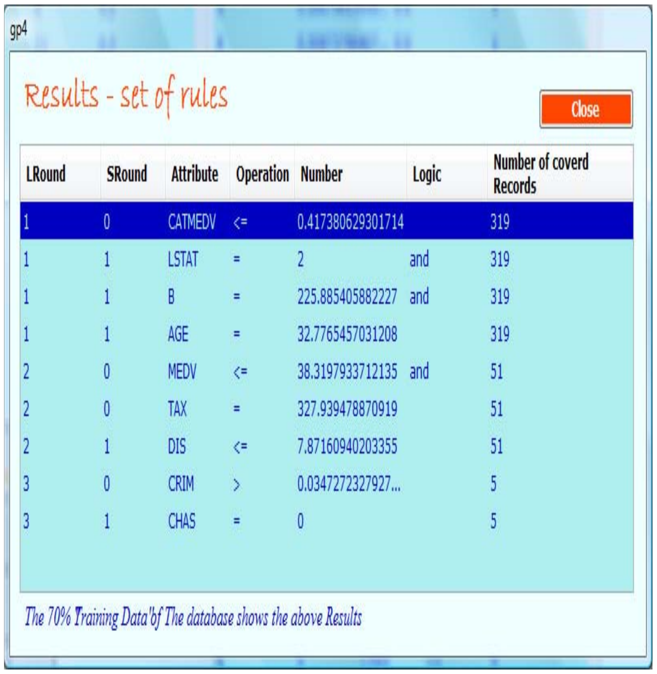
TABLE .2 THE ACCURACY OF CLASSIFICATION BY VARIOUS GP SYSTEMS

System	Accuracy
LOGENPRO	91.04%
GPSQL Miner	95.04%
Neural Net	96.7%
GP-SQL Mining Tool	**96.8%**

The experimental set of rules results in one run through GP-SQL Miner tool in this dataset is showed Fig. 4. The results of the data mining in the Boston housing database reveal the effectiveness of the proposed GP system. To start with, the system was able to apply the genetic operators; thus it was able to replicate programs without changes for the next generation (reproduction), select any two parent programs to generate new offspring (crossover) and randomly select non-terminal points from parent programs. Furthermore, it allowed diversity in the population of programs to prevent an early derivation of algorithm solution. The results of the experiment show that the tool provided excellent results. In fact, the fitness of 96 % is highly accurate.

V CONCLUSION AND FUTURE WORK

In this paper, we have explored the idea of combining data mining and genetic programming. Thus, data mining is a process of retrieving useful information from electronic databases, or rather; it is a process of knowledge discovery. This study was focused on highlighting the benefits that genetic programming can add in data mining. It did not examine the GP application in classification in a real work situation, which is vital if the tool needs to be applied in commercial settings. The achieved tests showed interesting and encouraging results in for data classification with a high efficiency rate. A subset of the obtained results is shown Fig. 4. On the other hand, GP is a technique that employs the evolutionary concepts to generate computer programs for use in classification task. Data mining involves a number of tasks, but more emphasis has been put on the classification task. Genetic programming is a based on genetic algorithms that use evolutionary concepts in the data mining process. It involves genetics operators such reproduction, mutation, and crossover. The technique has developed from the simple linear-based GP to cellular type of GP. We proposed a GP-based system that uses a heuristic approach in the induction of classifiers.



LRound	SRound	Attribute	Operation	Number	Logic	Number of coverd Records
1	0	CATMEDV	<=	0.417380629301714		319
1	1	LSTAT	=	2	and	319
1	1	B	=	225.885405882227	and	319
1	1	AGE	=	32.7765457031208		319
2	0	MEDV	<=	38.3197933712135	and	51
2	0	TAX	=	327.939478870919		51
2	1	DIS	<=	7.87160940203355		51
3	0	CRIM	>	0.0347272327927...		5
3	1	CHAS	=	0		5

The 70% Training Data'bf The database shows the above Results

Figure 4. Set of results (predicted Models)

A number of literatures were examined and studied. It is shown that GP techniques are more superior to traditional classification techniques in data mining. It is possible to compare same type attributes within a dataset to generate a reduced number of effective rules. Several GP-based techniques have been proposed. These range from a concept mapping technique that evaluates individual fitness through booleanized attributes and token completion, cellular GP, linear GP, to GPSQL Miner tool that uses the SQL grammar for classification. A variant GP approach uses the C-programming language in representation of genetic programs. Our system utilizes the SQL grammar; this is significant as it makes it easy to evaluate data stored in the database. In addition, the design employs heuristic measures such as cross validation, fitness function, elitism, and genetic operators like reproduction, and cross over to produce more favorable results.

In the reproduction process, individuals are copied exactly as they appear, while in mutation and crossover there changes in the offspring program. However, crossover involves exchange of properties between different programs, while mutation exchange can happen within the program. The fitness function was important in determining the accuracy of the system. Elitism prevents the loss of chromosomes with good quality. The proposed system validated the assumption that GP is an effective technique for data mining classification. It was designed to reduce the search space considerably making it easy to induce classifiers and classify the data. A tool was developed through VB.NET programming language, and an Ms Access relational database was used to store data. The results showed that the system is promising in regards to its application in classification of data in the data mining process It generated reliable results that are highly accurate when a large database having a lot of noise was used. The system was further validated against other common GP-based approaches for data mining classification including GP-Knn and GPSQL Miner. The level of accuracy for the system is higher than that achieved by both the GPSQL Miner and GP-Knn. Furthermore, the literature revealed that the various methods integrated into the system are beneficial in maintain the best chromosomes, reducing the search space, and generating the most suitable program for classification. The system maintains the advantages of using the SQL grammar in manipulating the database. However, the use of a high-level programming language increased the execution time when compared to a system that is based on a low-level language. It is recommended to conduct further tests in a commercial setup. This study was focused on highlighting the benefits that genetic programming can add in data mining. Therefore, future works will involve investigating the effectiveness of the system and such a GP approach on data mining classification on more databases, both small and large, and which comprise real data.

REFERENCES

[1] W. Banzhaf, P. Nordin, R. Keller, and F. Francone, "Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications", Morgan Kaufmann, 1998.

[2] S. Bhattachariya, O. Pictet and G. Zumbach, "Knowledge-intensive genetic discovery in foreign exchange markets". IEEE Transaction on evolutionary Computation, Vol 6 (2), April 2002, pp. 169-181.

[3] E. Carreno, G. Leguizamon, and N. Wagner, "Evolution of classification rules for comprehensible knowledge discovery." In proceedings of Evolutionary Computation Congress CEC 2007, pp. 1261-1268.

[4] P. Espejo, S. Ventura and F. Herrera, "A survey on the application of genetic programming to classification." IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol 40(2), March 2010. pp. 121-144.

[5] G. Folino, C. Pizzuti and G. Spezzano, "GP ensembles for large-scale data classification." IEEE Transaction on Evolutionary Computation, Vol 10(5), Oct 2006, pp. 604–616.

[6] Y. Ishida, and A. Pozo, "GPSQL Miner: SQL-grammar genetic programming in data mining." CEC'02 Proceedings of the Evolutionary Computation Vol 02, May 2002, pp. 1226–1231.

[7] S. Muharram and G. Smith, "Evolutionary constructive induction". IEEE Transactions on Knowledge and Data Engineering, Vol 17(11), Nov 2005, pp. 1518-1528.

[8] S. Sakprasat, and M. Sinclair, "Classification rule mining for automatic credit approval using genetic programming." In proceedings of Evolutionary Computation Congress CEC 2007, Sept 2007, pp. 548-555.

[9] M. Saraee, and R. Sadjay, "Optimizing classification techniques using genetic programming approach." In proceedings of INMIC 2008, Dec 2008, pp. 345-348.

[10] P. Tan, M. Steinbach, and V. Kumar, "Introduction to Data Mining." Addison Wesley Editions, 2006.

[11] H. Tan and E. Frank, "Data mining: Practical Machine Learning Tools and Techniques" Second Edition, Morgan Kaufman, 2005.