

Rule Extraction Algorithms on Type-2 Diabetes Data

Ramalingaswamy Cheruku, Pradeep Kumar Nalluri, Naga Sanketh Vysyaraju, Gopi Krishna Yogeshwar Gavuji, Balaji Sai Charan Jalukuru, Veeraiah Talagondapati

ABSTRACT-- Diabetes is a metabolic disorder in which body produces insufficient insulin or no insulin. As a result blood sugar levels in the body increases and causes severe complications in the long run. Medical decision support systems play a crucial role in identifying the diabetes in the early stages. This helps the doctors to provide better diagnostics to increase patient lifespan. In the literature there are many such decision support systems available. In this paper we are presenting the performance of such systems and this helps the researchers working in this field.

Keywords: Survey, Diabetes, Support Systems

1. INTRODUCTION

Tracking system are becoming essential now a

Diabetes is a most common endocrine system disorder where in, the glucose levels in blood consistently stay high or above normal conditions leading to diabetes. Human body uses glucose present in blood or also known as blood sugar for energy which is a main source of energy. The glucose levels in the blood come from the food taken. The food taken is converted in to glucose which is then utilized by body for energy. Insulin, a hormone produced by pancreas plays an important role in regulating the amount of glucose in blood. It allows glucose from blood to get in to cells for energy. Irregularities in producing the amounts of insulin by pancreas leads to high levels of glucose in blood resulting in health problems especially diabetes.

In pre-diabetes stage we have a condition in which sugar (glucose) levels in blood are higher than normal but are not high enough to consider them as diabetes. There are two types of diabetes namely type-1 and type-2. In type-1 diabetes insulin is not produced at all. So the glucose levels are very high and people suffering from type-1 diabetes have to take insulin injections regularly to avoid any drastic effects. It is the most severe form of diabetes and is usually diagnosed in kids and adults although it can occur at any age. Type-2 diabetes is the most common form of diabetes wherein the body does not produce enough amounts of insulin in the body, gradually leading to higher amount of glucose in the body. 95% of people suffering from diabetes are of type 2. Even

though this type can occur in people of all ages, it is most prevalent in middle aged people (above 35 yrs.).

Diabetes is the most common disease. According to a survey in 2015 by the American Diabetes Association [3], around 30.2 million (9.4% population) people are suffering from diabetes in US. Out of these, 7.2 million people were undiagnosed. Usually the symptoms of type-1 disease appear quickly within weeks of its origin. But the symptoms of type-2 disease are very hard to identify. People suffering from type-2 diabetes identify the symptoms only after long time when the disease causes long term damage. This is a very important problem, identify diabetes at its initial stage and diagnose it to prevent its long term damage. Many classifiers using rule miners are available in decision making and classification of the disease. These results aid doctors for better diagnosis. This paper gives a brief overview of major rule mining classifiers available which will help the research in the medical field. [1,2,4]

2. PRELIMINARIES

2.1. Description of dataset

As mentioned, the importance of getting the knowledge of the Diabetes is a paramount task, in this survey we are observing the Pima Indians Diabetes (PID) Dataset which is sourced from UCI machine learning repository. This dataset have 8 feature attributes and 1 decision attribute. There are two classes positive (label of the class is 1) and negative (label of the class is 0). Their count in the dataset are 268 and 500 respectively making a total of 768 data vectors.

The features of the dataset are shown in the following table

Feature List	
Feature Number in the dataset	Name of the feature
Feature 1	Number of times the patient is pregnant
Feature 2	Concentration of Glucose and Plasma
Feature 3	Diastolic Blood Pressure
Feature 4	Triceps skin fold Thickness
Feature 5	Serum insulin
Feature 6	Body mass index
Feature 7	Diabetes pedigree function
Feature 8	Age
Feature 9	Label of the Class

Table 1. Different Features in PIMA Indian Diabetes dataset

Revised Manuscript Received on February 11, 2019.

Ramalingaswamy Cheruku, Department of Computer Science and Engineering, Mahindra Ecole Centrale, Hyderabad, Telangana, India. (E-mail: ramalingaswamy.cheruku@mechyd.ac.in)

Pradeep Kumar Nalluri, Department of Computer Science and Engineering, Mahindra Ecole Centrale, Hyderabad, Telangana, India.

Naga Sanketh Vysyaraju, Department of Computer Science and Engineering, Mahindra Ecole Centrale, Hyderabad, Telangana, India.

Gopi Krishna Yogeshwar Gavuji, Department of Computer Science and Engineering, Mahindra Ecole Centrale, Hyderabad, Telangana, India.

Balaji Sai Charan Jalukuru Department of Computer Science and Engineering, Mahindra Ecole Centrale, Hyderabad, Telangana, India.

Veeraiah Talagondapati, Department of Computer Science and Engineering, Mahindra Ecole Centrale, Hyderabad, Telangana, India.

2.2 Model for rule extraction using Q-Learning, Fuzzy ARTMAP (FAM) and Genetic Algorithms (GA)-QFAM-GA

QFAM-GA[5] is a Two-Stage hybrid model used for the classification of data and extraction of rules.

Stage 1: Q-Learning with Fuzzy ARTMAP (QFAM) with pruning

The training of model is same as in FAM, the only difference is every node is given a Q-value which is based on the Q-learning a type of Reinforcement Learning Technique.

Learning: If we consider the learning phase of this model (QFAM), the winner that is selected in the recognition layer of the fuzzy ART(f_a^2) may result in inaccurate predictions, this leads to match-tracking in case of wrong predictions and learning in case of correct predictions. So, the winning node in f_a^2 is rewarded or penalized based on the prediction outcome and the Q-value is updated using the following expression

$$Q_t(j) = Q_{t-1}(j) + \xi[r_t(j) + \gamma \text{vig}(j)] \quad (1)$$

where γ belong to $[0,1]$ is the Discount Factor

ξ belongs to $[0, 1]$: Learning Rate

$\text{vig}(j)$: winning node's vigilance value of f_a^2

$r(j)$ belongs to $\{0,1\}$ reinforcement signal defined as:

$$r(j) = \begin{cases} 0 & \text{if learning} \\ 0 & \text{if match - tracking} \end{cases}$$

Prediction: This is done with the help of Fuzzy Art Map, First a choice function (T_j) is employed, which is as follows:

$$T_j = \frac{|A \wedge w_j^a|}{\alpha + |w_j^a|} \quad (2)$$

here $w_j^a = (w_{j1}^a, \dots, w_{j2M}^a)$ is a weight vector of the node j in f_a^2 and w_j is generated with $A = (a, 1-a) \equiv (a_1, \dots, a_m, 1-a_1, \dots, 1-a_m)$ here a is input vector and then a vigilance test shown in eq[4] is imposed on to these nodes of f_a^2 . The nodes that passes the vigilance test are recognized to participate in the next selection process.

In this process strength of each selected node is calculated by:

$$\text{strength}(j) = \lambda T(j) + (1 - \lambda)Q(j) \quad (3)$$

where λ belongs to $[0,1]$ is the weighting factor

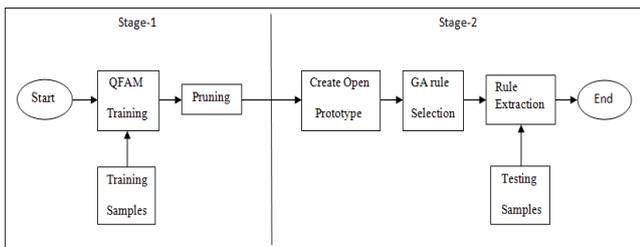


Fig.1. QFAM two-stage model

Pruning: This is done to reduce the size of population. Nodes with lower Q-value compared to a threshold are dropped in this phase. The remaining nodes are used for rule extraction and Q-value is used for if-then rule confidence extraction.

Stage 2: Extraction of rule with the aid of Genetic Algorithm (GA)

This is done to get a precise and accurate rule set for classifying. The remaining nodes after the pruning in step 1 are taken and are used for creating the open prototypes which has the don't care feature.

Creating Open Prototypes: Dimensions of open prototypes in QFAM are same as in the input features. If the number of dimensions is high, then rule antecedent is more. Then extracted rules get complicated, so the concept of don't care is used to tackle the problem. The feasible rules for search space are huge so GA is the effective way to tackle the problem. Each node can create $2^d - 2$ open prototypes, the high and the low of each dimension are set to 1 and 0 respectively. These values which are appropriate to the don't care feature are used for the calculation of the choice function and vigilance test, which is given by

$$\frac{|A \wedge w_j^a|}{|A|} \geq \rho_a \quad (4)$$

Here ρ_a is the vigilance parameter of ART_a , T_j is the choice function which is set to zero if this test is not followed.

The main goal is to find a very accurate rule set for the classification, here GA is used to extract the most accurate rules for the classification problem. The chromosome used in GA is

$$D_1^1, D_2^1, \dots, D_d^1, D_1^2, D_2^2, \dots, D_d^2, \dots, D_1^p, D_2^p, \dots, D_d^p$$

Where p is the total number of prototypes and d is the total number of features for each prototype

$$D_d^p = \begin{cases} 0 & \text{for don't care features} \\ 1 & \text{for other features} \end{cases}$$

Process of GA: The main aim of using GA in this model is to optimize the accuracy of the classification and to reduce the input features, this can be achieved by using the following fitness equation:

$$f(S) = W_{NCP} \cdot NCP(S) - W_s \cdot |S| \quad (5)$$

where $|S|$ is the total number of features in the dataset and $NCP(S)$ are the data samples which are classified correctly. W_s and W_{NCP} are the +ve weights.

Initialization: Randomly generate initial population of size N_{pop}

Selection: Select $\frac{N_{pop}}{2}$ samples from the population of current generation. The selection probability of each candidate is given by

$$P(S) = \frac{f(S) - f_{min}(\Psi)}{\sum_{S \in \Psi} f(S) - f_{min}(\Psi)} \quad (6)$$

here Ψ is the length of the population and

$$f_{min}(\Psi) = \min \langle f(S) | S \in \Psi \rangle$$

Crossover: Randomly choose a bit position (S_r) to perform crossover among the chosen pair depending on the crossover probability.

Mutation: Mutation is applied to the strings that are generated in crossover using the mutation probability (P_m)

$$S_r = 1 \rightarrow S_r = -1 \text{ and } S_r = -1 \rightarrow S_r = 1 \text{ with}$$

probability $P_m(1 \rightarrow 1)$ and $P_m(-1 \rightarrow 1)$ respectively

Elitism strategy: If the best value in the current generation is worse than the best value in the previous generation then the worst value string is replaced by the best value string in the previous generation

Termination: If termination condition is satisfied then terminate the process, else go back to step1.

Rule Extraction:

Every prototype can be taken as a fuzzy rule. For developing linguistic based rule extraction, quantization of input features is done. The level of Quantization Q gives the total number of fuzzy partitions that fall in the quantized level. The interval [0, 1] is splitted into Q different partitions and the round of method is applied for quantization.

$$V_q = \frac{q-1}{Q-1} \quad (7)$$

Where $q = 1, 2, \dots, Q$, V_q is the antecedent value

In this model the rules are as follows:

Rule (R_j) is given by

IF $X_{p1} = V_q$ and $\dots X_{pn} = V_q$

THEN X_p belongs to class C_j with $Q_{value} = Q_{values}(j)$

where j denotes the prototype post the pruning, X_p is the data vector of n dimensions and the Q-value is calculated using eq(1). Thus, the rules are extracted using QFAM - ARTMAP.

2.3 Ant Colony Optimization (ACO)

In medical domain, to extract patients data which aids in diagnosis of the disease classification systems have been used widely. Fuzzy logic is a well known solution for classification problems. It improves the decision making systems and allows overlapping class definitions and their capabilities to handle vagueness and uncertainties. Several methods were produced to generate the fuzzy rules. Some include using Genetic based algorithms to generate fuzzy rules, some others used multiple Genetic based algorithms to produce fuzzy rules in which Ant Colony Optimization (ACO)[6] was used for local search procedure which resulted in improving the final classification. ACO was first introduced in 2002 for classification which was called the ANTMINER. Artificial ants are deployed for the construction of set of fuzzy rules which produced promising results. Another version of ANTMINER was introduced which instead of using entropy as a heuristic information, used a simpler function which reduced the computation overhead. It was called ANTMINER2. Later 2 more versions of ANTMINER were introduced with modifications to the previous versions called ANTMINER3 and ANTMINER+ respectively. This subsection presents another version of ANTMINER called FCS-ANTMINER.

Artificial ants are utilized by FCS-ANTMINER to explore the space, generate fuzzy rules gradually. Directed acyclic graph is the environment in this version of ANTMINER. In the older versions of the ANTMINER each ant will start making a rule and it tries to select the best among the rules available. This results in ants competing with each other to select high quality rules. This results in increased competition between the ants to select the best rule. Increased competition makes some ants to pick some high quality rules while others will not able to find correct

paths to choose high quality rules. Hence the output contains both low accuracy and high accuracy rules. As a result, the final classification system may not be able to significantly classify the number of input patterns. One of the reasons for the above problem is that the ants do not co-operate with each other. FCS-ANTMINER tries to balance the competition along with the cooperation between the ants in a manner that the ants are motivated to help the colony in building the rules accurate. In the proposed method, according to a parameter the succeeding ants modify a rule constructed by their preceding ants. The value of the parameter decides the number of terms that the ant can modify in each of the iteration. This improves the classification system accuracy along with helping ants in learning high quality rules.

The classification rate of the proposed method was 84.24% with accuracy of 0.8413. The results show that the FCS-ANTMINER using 10-fold cross validation has better accuracy and classification rate than that of previously established and well known ANTMINERS.

2.4 Artificial Bee Colony (ABC)

The ABC [7] is a swarm based optimization algorithm. The Artificial bees colony contains three groups of bees called employed bees, scout bees, onlooker bees. Employed bees are the ones with the food sources, scout bees search for the food sources randomly, onlooker bees look at the dance of the employed bees for information about food source and choose food source according to a probability. Initially all food sources are discovered by scout bees and then the nectar of food sources are exploited by employed and onlooker bees. The employed bees bring loads of nectar to the hive and become scout bees when they are exhausted. When they bring back nectar to hive they dance to show the where the food source is and the onlooker bees select the food source by the dance of the employed bees.. In ABC algorithm, the position of the food source represents the possible solution and the amount of nectar in the food source represents the solution fitness .

2.5 Modified ABC fuzzy classifier learning

There is some insufficiency in ABC, because ABC works well for exploration but lacks in exploitation of the food source. To improve this aspect a new algorithm was proposed called Modified ABC [7] which improves local search and convergence characteristics of ABC. Here we first create an initial population as it is a crucial task in evolutionary algorithms and can affect the speed of convergence along with the final quality of output. In this modified ABC every onlooker bee produces a new solution near to its present solution and the generated solution would be better or worse depending on the used fitness function. If the generated solution is better, then that is used for next iteration. If it is worse may be due to the lack of variety of solutions, a new searching mechanism is used which helps to search for wide varieties of solutions. This is adopted by blended crossover operator of GA into ABC for improving the exploitation process.



2.6 Particle Swarm Optimization (PSO) with NEFCLASS

One of the most powerful system of organizing knowledge and data is through classification systems. Now a day's classification systems are used in various fields to identify, sort and to create models of knowledge and one such field is medical diagnosis. One of the many classification approaches that can be used in diagnosis are neuro-fuzzy classifiers, which aims at creating fuzzy classification rules and data sets. It can be defined as a system that is trained with some neural network algorithms, which generates a more robust, efficient system with good classification accuracy. One of the simplest models of Neuro-fuzzy classification models on which we are discussing in this section is NEFCLASS, which learns rules and data sets by heuristics from the training data for creating classifiers. The primary purpose of it is to create readable classifiers by taking care that the fuzzy data sets cannot be arbitrarily modified by learning and in addition constraints can be placed to make sure that the fuzzy sets fit to their corresponding language labels after learning. A new hybrid approach has been proposed for this model using PSO [8]. This PSO is used for adjusting the membership parameters.

The PSO is an Evolutionary technique, which is computationally less expensive for a given size of network topology. The main advantage of this algorithm is that unlike other GA (Genetic Algorithms) it does not have operators for evolution such as mutation, crossover and moreover it has less parameters to adjust. Now, the similarities of this with the other evolutionary algorithms are that it also has a set of candidate solutions called particles, which are spread over the search space. But here each particle is associated with a particular position and velocity with which it moves across the solution space over generations called iterations. Each particle is also associated with a memory to store the best position (highest fitness value) that it has ever visited. So, it moves towards the current best position. It also stores the Global best position that any particle has ever visited. The particles update their velocities and positions based on two equations. They are,

$$V_i^{k+1} = \gamma v_i^k + c_1 r_1 (P_{besti}^k - X_i^k) + c_2 r_2 (P_{Gbesti}^k - X_i^k) \tag{8}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{9}$$

Here,

V_i^k is the i^{th} particle velocity in the k^{th} iteration

X_i^k - the position of the particle.

C_1, C_2 - positive constants of acceleration, which helps in keeping the balance between the individual particles and social behavior.

r_1, r_2 - two random number between 0 and 1 are added to include randomness into the model.

P_{besti}^k - the best position of the particle i , achieved from its own experience in the k^{th} iteration

P_{Gbesti}^k - the global best position of the particle from overall experience of all particles in the k^{th} iteration

γ - the inertia weight for maintaining the balance between the global and local search abilities.

The core of the PSO is to find the optimum position. The method (PSO with NEFCLASS) unlike the normal NEFCLASS, which uses simple heuristics to adjust the

fuzzy sets replaces the fuzzy subsets with the particles created in PSO algorithm. After each iteration the error is calculated for each particle. The minimum of all those is the global best, to which all the particles approach by calculating their new positions. Once the process is completed the best particle is retrieved along with its parameters and replaced with the initial subset medium.

In this study, two models were compared, they are :
Classical NEFCLASS

PSO - NEFCLASS

The experiments on the above two models showed that the model developed by them in this study does more accurate diagnosis than the classical one with substantial improvement.

2.7 Parliamentary Optimization Algorithm (POA)

This algorithm is used for automatic classification rule discovery for datasets with numerical attributes. The POA [9] algorithm is not used in any real applications yet but the results are very promising compared to other methods. The objective function used in this method is very flexible and different objectives can be added easily.

In the process of classification, classification rules are identified using training data set and they are used to identify the class of an object whose class is unknown. Many classification methods have been established such as decision trees, support vector machines (SVM), K-nearest neighbors, neural networks etc. But all these methods are like a black box and obtaining accurate rules is more important in classification tasks of data mining. Obtaining accurate rules is even more complex in case of numerical attributes. In other classification processes, the data set is modified as a priori process and so modified data set is found in process of classification. Often discretization or fuzzification is used to modify the data. The POA method finds numerical attribute intervals automatically without any priori process.

Metaheuristic methods are often preferred in non-linear and multi-model real world optimizations with complex constraints. The POA method is metaheuristic. Many metaheuristic optimizations are bio, social, physical, chemically inspired processes. Parliamentary Optimization Algorithm is socially inspired by the competition within parties in a parliament. The various steps involved in the algorithm are:

- 1) Population initialization and partitioning
- 2) Intra group competition
- 3) Inter group competition
- 4) Termination

Population initialization and partitioning: The initial population consists of M individuals divided into L groups. Within each group θ best individuals are identified and classified as leaders and the remaining as regular members in the group.

Intra group competition: Regular members are biased towards candidates. New position of the candidate and the power of group is calculated.



The new position of the candidate is calculated according to the below formula

$$p' = p_0 + \pi \left\{ \frac{\sum_{i=1}^{\theta} (p_i - p_0) \cdot f(p_i)}{\sum_{i=1}^{\theta} f(p_i)} \right\} \quad (10)$$

Here π is a random number between 0.5 and 2. f is the fitness function. p_0 is

0 the current position of the regular member, p is the new position of the regular member and p_i is a candidate member.

The power of each group is calculated according to the below formula:

$$power_i = \frac{a \times average(Q_i) + b \times average(R_i)}{a + b}; a > b \quad (11)$$

Where a, b are the weighing constants which are predetermined, Q_i and R_i are the fitness values.

Inter Group Competition: A random number is being generated. If it is less than the probability value p_m , then λ most powerful groups are merged. Likewise, a random number is generated and if it is less than the probability p_d then Y weaker groups are eliminated.

Termination: If maximum number of iterations are reached or if the fitness value doesn't change across few iterations then the algorithm terminates. At the end of the algorithm, the best member of the group is the best solution for the optimization problem.

The fitness value of each member of the group is calculated and θ of them are taken as leaders and the remaining as regular candidates. Formula for calculating the fitness function is :

$$fitness = w_j X \frac{TP}{TP+FN} X \frac{TN}{TN+FP} - w_2 X comprehensibility + w_3 X \frac{TP}{TP+FP} \pm w_4 X interval rate \quad (12)$$

$$comprehensibility = \frac{\text{number of attributes in the left side of rule} - 1}{\text{number of all attributes}} \quad (13)$$

$$interval rate = \sum_{i=1}^n \frac{Ub_i - Lb_i}{Att_{imax} - Att_{imin}}$$

TP = number of true positives (the actual class of test is positive and POA predicts it correctly as positive)

TN = number of true negatives (the actual class of test is negative and POA predicts it correctly as negative)

FP = number of false positives (the actual class of test is negative and POA predicts it wrongly as positive)

FN = number of false negatives (the actual class of test is positive and POA predicts it wrongly as negative) w_1, w_2, w_3, w_4 are weighting factors

2.8 SM Rule Miner

Spider Monkey Optimization (SMO) [10] is based on behavior of spider monkeys it falls under the category of fission and fusion social structure. Spider monkeys live in groups while searching for food in groups, they divide themselves in to subgroups to minimize the competition between them. Normally a female member will lead in their respective groups and searches for food. In this the female is responsible for searching food sources, there will be a global

leader who checks for the food whether it is sufficient for the group or not. If not it divides the group further in to small subgroups, which may have (3 to 8) members in a subgroup and these subgroups are also led by a female member of their respective groups also known as local leader, they are also responsible for planning an efficient searching route each day.

Each subgroup leaders communicate with their group leaders to maintain social bonds and boundaries. They check the distance every time they find a food source, each member updates their position according to their distance. Here the local leader decides the best position within group. Each member searches for food in different directions. Later the global leader updates it's position according to the local leader's positions and it takes the best position out of all the local leader's positions. This happens continuously, initially SMO generates the population and the groups are divided in to subgroups, then the local leaders find the best position from the group members positions from the food sources. After this the global leader finds the best position from the local leaders best positions.

Each spider monkey updates its position as

$$SM_i = SM_{ij} + .. + SM_{Dj}$$

SM_i represents the i^{th} candidate rule.

Every SM updates its position in above equation as.

$SM_{ij} = SM_{LBj} + U(0,1) * (SM_{UBj} - SM_{LBj})$ (During Initialization phase)

SM_{LBj}, SM_{UBj} are lower and upper bounds of the j^{th} attribute

$SM_{newij} = SM_{ij} + U(0,1) * (LL_{kj} - SM_{ij}) + U(-1,1) * (SM_{rj} - SM_{ij})$ (During

Local leader phase)

LL_{kj} is the k^{th} local group leader position at the j^{th} attribute

SM_{rj} represents the r^{th} attribute of the r^{th} SM

$SM_{newij} = SM_{ij} + U(0,1) * (GL_j - SM_{ij}) + U(-1,1) * (SM_{rj} - SM_{ij})$ (During

Global leader phase)

Where GL_j is j dimension of global leader randomly chosen from 1 to D indexes

3. GRAPHS

The comparison of accuracies, sensitivities, specificities of different models can be viewed in figures-2,3 ,4 respectively.

Accuracy: It is a measure that is used to evaluate the performance of an algorithm, which is defined as how correctly the model has predicted the outcomes of the test samples in the test dataset.

$$Accuracy\% = \frac{\text{Number of correctly predicted samples}}{\text{Total number of samples in the test dataset}} \times 100 \quad (15)$$



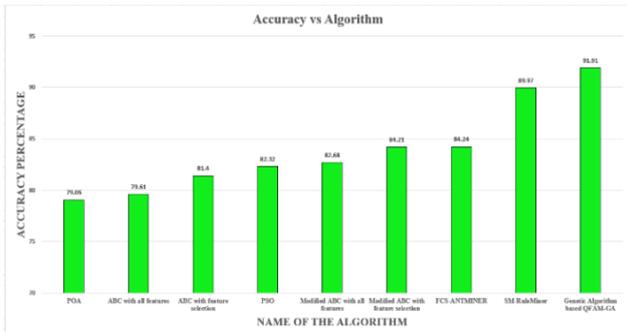


Fig.2. Accuracies vs Algorithm

Sensitivity: It is defined as the extent of correctness of the model in predicting the positive samples of the test dataset.

$$Sensitivity\% = \frac{\text{Number of correctly classified positives}}{\text{Total number of positives in the test dataset}} \times 100$$

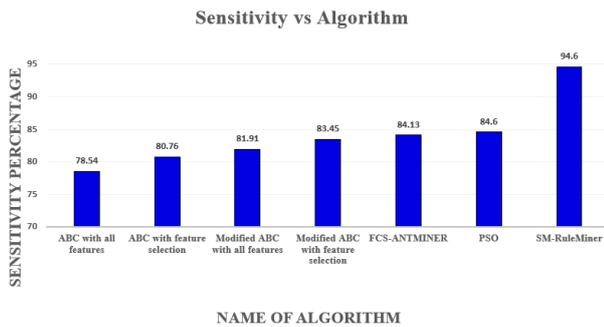


Fig.3. Sensitivity vs Algorithm

Specificity: It is defined as the extent of correctness of the model in rejecting the negative samples that belong to the test dataset.

$$Specificity\% = \frac{\text{Number of correctly rejected negative samples}}{\text{Total number of negative samples in the test dataset}} \times 100$$

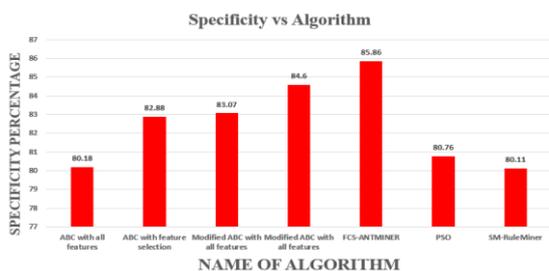


Fig.4. Specificity vs Algorithm

Figure-5 depicts the development of the algorithms over the years.

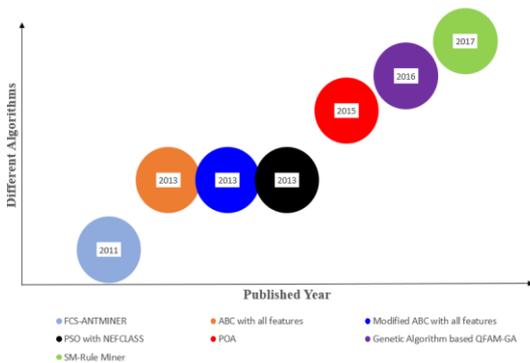


Fig.5. Year vs Algorithm

4. COMPARATIVE ANALYSIS & RESULTS

The below table 2 illustrates accuracy, sensitivity, specificity measures of different algorithms discussed in the previous sections.

Algorithm	Accuracy (%)	Sensitivity (%)	Specificity (%)	Author (year) [Refs.]
FCS-ANTMINER	84.24	84.13	85.86	M.F Ganji et al.(2011)[6]
Modified ABC + all dimensions	82.68	81.91	83.07	F.Beloufa et al.(2013)[7]
Modified ABC +Dimensionality reduction	84.21	83.45	84.60	F.Beloufa et al.(2013)[7]
ABC + all dimensions	79.61	78.54	80.18	F.Beloufa et al.(2013)[7]
ABC +Dimensionality reduction	81.40	80.76	82.88	F.Beloufa et al.(2013)[7]
PSO	82.32	84.60	80.76	M.E.H Dabo et al.(2013)[8]
SM-Rule Miner	89.87	94.60	80.11	C. Ramalingaswamy et al.(2017)[10]
POA	79.06	-	-	S.Kizilolok et al.(2015)[9]
QFAM-GA	91.91	-	-	F.Popurpanah et.al.(2016)[5]

Table 2. State-of-the-art rule mining algorithms performance comparison. “-“ denotes value is not available.

From the table 2 we infer the following points:

- In terms of accuracy, the algorithm QFAM-GA outperforms all the other algorithms by a fractional margin because of effective feature selection that happens during the Genetic Algorithm phase of QFAM-GA.
- In terms of sensitivity the SM-Rule miner algorithm outperforms all the other algorithms significantly.
- In terms of specificity ACO dominates all the other algorithms.

5. CONCLUSION

Rule-based classification algorithms play a vital role in the medical decision support systems. Hence, in this paper we reviewed dataset rule mining algorithms for diabetes classification. We observed from comparative analysis that in terms of accuracy QFAM-GA and in terms of sensitivity SM rule miner algorithm and in case of specificity ACO rule mining algorithm outperformed all others on the pima diabetes dataset. Having seen the specificities, sensitivities and accuracies of the different well established algorithms,



the best among them can be chosen based on the requirement by going through the values of different metrics shown in table

ACKNOWLEDGEMENT

Authors would like to thank Director of Mahindra Ecole Centrale, Hyderabad for providing constant support.

REFERENCES

1. About Diabetes from National Institute of Diabetes and Digestive and Kidney Diseases and can be found at <https://www.niddk.nih.gov/health-information/diabetes/overview/what-is-diabetes>
2. About Diabetes from Research Institute Foundation and can be found at <https://www.diabetesresearch.org/what-is-diabetes>
3. Statistics About Diabetes from American Diabetes Association and can be found at <http://www.diabetes.org/diabetes-basics/statistics/>
4. Early Diabetes Symptoms from n WebMD and can be found at <https://www.webmd.com/diabetes/guide/understanding-diabetes-symptoms#1>
5. Pourpanah, Farhad, Chee Peng Lim, and Junita Mohamad Saleh. "A hybrid model of fuzzy ARTMAP and genetic algorithm for data classification and rule extraction." *Expert Systems with Applications* 49 (2016): 74-85.
6. Ganji, Mostafa Fathi, and Mohammad Saniee Abadeh. "A fuzzy classification system based on Ant Colony Optimization for diabetes disease diagnosis." *Expert Systems with Applications* 38.12 (2011): 14650-14659.
7. Beloufa, Fayssal, and Mohammed Amine Chikh. "Design of fuzzy classifier for diabetes disease using Modified Artificial Bee Colony algorithm." *Computer methods and programs in biomedicine* 112.1 (2013): 92-103.
8. Daho, Mostafa El Habib, et al. "Recognition of diabetes disease using a new hybrid learning algorithm for nefclass." *Systems, Signal Processing and their Applications (WoSSPA), 2013 8th International Workshop on. IEEE, 2013.*
9. Kiziloluk, Soner, and Bilal Alatas. "Automatic mining of numerical classification rules with parliamentary optimization algorithm." *Advances in Electrical and Computer Engineering* 15.4 (2015): 17-24.
10. Cheruku, Ramalingaswamy, Damodar Reddy Edla, and Venkatanareshbabu Kuppili. "SM-RuleMiner: Spider monkey based rule miner using novel fitness function for diabetes classification." *Computers in biology and medicine* 81 (2017): 79-92.