Cihan University, First International Scientific conference

Research Article

Design and Implementation of a Meta Knowledge System (MKS)

Samaher Hussein Ali[a], Ahmed Patel[b,c] and Ehab Hussein Ali[d]

[a] Department of Information Network, Faculty of Information Technology, University of Babylon, Babylon, (00964), Iraq

[b] School of Computer Science, Centre of Software Technology and Management (SOFTAM), Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor Darul Ehsan, Malaysia

[c] Visiting Professor, School of Computing and Information Systems, Faculty of Science, Engineering and Computing, Kingston University, Kingston upon Thames KT1 2EE, United Kingdom

[d] Department of Electrical Engineering, College of Engineering, University of Babylon, Babylon, Iraq (00964)

Email: [a]Samaher @itnet.uobabylon.edu.iq, [b,c]whinchat2010@gmail.com, and [d]Eng.ehab_hussein1979@yahoo.com

Abstract

The purpose of this research is to develop a self-organizing network. The network initially has only input neurons. During the training process, neurons are selected from a pool of candidates and added to the hidden layers. These are the polynomial and hyperbolic functions which include eleven polynomial functions. During the final stage, the proposed system performs analysis for each model based on five error predicating measures including Maximum error, RMSE, MSE, MAE and MAPE. Through experiments we found that all the huge databases have fixed behaviors, the best model generated by linear of three variable functions and worst model generated by cubic of one variable or quadratic function related to the polynomial models. The best model generated by tanh of one variable function and the worst model generated by more than one other function is related to the hyperbolic models. Most of the small databases have unstable behavior, the best model is generated by linear of three and two variables or quadratic of two variable functions and the worst model is generated by cubic of one variable or quadratic functions related to the polynomial models. Finally, the best model generated by inverse of tanh of one variable function and the worst model generated by sinh or cosh functions are related to the hyperbolic models.

Keywords: Knowledge management applications, Meta knowledge, Self organization network, Hyperbolic Functions, Polynomial Functions.

1. INTRODUCTION

The basis of research of this paper is to design and implement a Meta Knowledge (i.e., knowledge about knowledge) system, that is, extraction of new knowledge (i.e., mathematical models) from the original knowledge (i.e., classified rules).

This work combines the advantages of both the Knowledge discovery algorithms and developing self- organizing networks to satisfy the concept of a Meta knowledge system

Initial focus is to develop self -organizing networks. The network begins with only input neurons. During the training process, neurons are selected from a pool of candidates and added to the hidden layers. The development of the network focuses on using different types of polynomial and hyperbolic functions to train the network.

In developing the self -organization network, the connections between the neurons in the network are not fixed but rather are selected during training to optimize the network. The number of layers in the network also is selected automatically to produce maximum accuracy without over fitting.

2. THE MAIN STAGES OF A META KNOWLEDGE SYSTEM

We summarize the main stages of the suggested system in this work by the following steps. In addition, Figure 1 shows the block diagram of the suggested system. The following stages are:

Stage 1: *Problem Reformualtion against latest Research Questions, Definition and Requirements Capture*

- Update study on Meta Knowlegde system, develop self-organizing networks, Ripper classifier, design multi mathmatical models based on the developed self-orgnizing network perspectives.
- Identify requirements and select modelling tools.

Stage 2: *Preprocessing Database.*

- Study the natural(i.e., the domain of database, type of their features, and their used) of the each selected DB
- Apply the normalization on the description features of that DB

Stage 3: *Specify and Design the Clssifier and develop the self- orgnizing neural network.*

- Apply the Ripper algorithm to generate a set of classification rules.
- Specify and Design a Self-orgnizing network based on replacing the activation function by the selected function (one of the polynomial or hyperbolic)
- Generation of the mathmatical model based on the selected functions

Stage 4: *Analysis and Evaluaion*

- Test the new framework, models against evaluation criteria we used five error predicating measures(Maximum error, RMSE, MSE, MAE, MAPE)
- Evaluate the framework
- Determine the best and worst models generated from this framework based on testing all the polynomial measures. *Also,* Best and worst model generated from this framework based on testing all the hyperbolic measures.

ALGORITHM OF HOW TO ACHIEVE THE MAIN STEPS IN MKS

Input: *Collection of Databases from different domains and is different in size and Natural,*

Output*: Best and Worst Model*

• ***Step1: Set the main parameters*** *to classify the algorithm (i.e., RIPPER), develop mathematical modeling algorithm (Self- organizing algorithm)*

• ***Step2: Generate the classify Rule Set***

- *For each select DB Є bank of databases*
  - *Call the procedure of RIPPER*
  - *Call the procedure for pruning the rule set*

• ***Step3: Generate the mathematical models***

Select one of the polynomial or hyperbolic functions

- *For each function related to polynomial functions do*
  - *Training the self –organizing neural network using the select function as activation function*
  - *Generate the mathematical mode*

End for

- *For each function related to hyperbolic functions do*
  - *Training the self –organizing neural network using the select function as activation function of it.*
  - *Generate the mathematical model*

End for

• ***Step4: Evaluation of the mathematical models***

- *For each model base on used polynomial functions do*
  - *Evaluation of the model base on five error predicating measures(Maximum error, RMSE, MSE, MAE and MAPE)*
  - *Save the results in bank1 and sort the results from the best to worst model base on the values of error predicating measures*

End for

- *For each model base on used hyperbolic functions do*
  - *Evaluation of the model base on five error predicatinge measures(Maximum error, RMSE, MSE, MAE and MAPE)*
  - *Save the results in bank2 and sort the results from the best to worst model base on the values of error predicatinge measures*

End for

  - *Display the results of bank1 and bank2 for the selected DB.*

End for

• ***Step5:*** *End MK Algorithm*

Procedure of Ripper

Input: *Selected Databases (i.e., database contains set of positive and negative samples*

Output*: Generate the classify Rule Set for all positive-"Pos" and Negative "Neg" samples*

If *E represents number of epochs in Ripper and determined in step1 of MKS* ***Then***

For each E

- *Rule Set = Optimize Rule Set(Rule set, Pos, Neg)*
- *Begin learning algorithm*
  - *Determine the length of the Rule Base (DL)*
  - *Make the Rule Set equal empty set*
  - *For each sample in Database do*
    - *Rule= learn Rule(Pos, Neg)*
    - *Add rule to Rule Set*
    - DL^* *=DL (RuleSet, Pos,Neg)*
    - *If* $DL^* > DL+45$
      - *Call the Pruning rule Procedure*
      - *Return Rule Set*

      End if
    - *If* $DL^* < DL$ *then*
      - $DL=DL^*$
      - *Delete samples covered from Pos and Neg*

      End if

    End for

  End for
  - *Return Rule Set*

End if.

Where, DL: description length of the rule base

The description length of a rule base = (the sum of the description lengths of all the rules in the rule base) + (the description of the samples not covered by rule base).

Table 1: Functions to Modeling the Classification Rules

Polynomial Functions		
Name	**Variables #**	**Functions**
Linear	**One**	F(Y)=P1+P2*Y1
Linear	**Two**	F(Y)=P1+P2*Y1+P3*Y2
Linear	**Three**	F(Y)=P1+P2*Y1+P3*Y2+P4*Y3
Quadratic	**One**	F(Y)=P1+P2*Y1+P3*Y1^2
Quadratic	**Two**	F(Y)=P1+P2*Y1+P3*Y1^2+P4*Y2 +P5*Y2^2+ P6*Y1*Y2
	One	F(Y)=P1+P2*Y1+P3*Y1^2+P4*Y1^3
Product	**Two**	F(Y)=P1+P2*Y1*Y2
Ratio	**Two**	F(Y)=P1+P2*(Y1/Y2)
Logistic	**One**	F(Y)=P1+P2/(1+exp(P3*(Y1-P4)))
Log	**One**	F(Y)=P1+P2*Log(Y1+P3)
Exponential	**One**	F(Y)=P1+P2*exp(P3*(Y1+P4))
Asymptotic	**One**	F(Y)=P1+P2/ (Y1+P3)
Hyperbolic Functions		
Name	**Variables #**	**Functions**
Sinh	**One**	$F(Y)=\frac{\exp(Y1)-\exp(-Y1)}{2}$
Cosh	**One**	$F(Y)=\frac{\exp(Y1)+\exp(-Y1)}{2}$
Tanh	**One**	$F(Y)=\frac{\exp(Y1)-\exp(-Y1)}{\exp(Y1)+\exp(-Y1)}$
$Sinh^{-1}$	**One**	$F(Y)=\frac{2}{\exp(Y1)-\exp(-Y1)}$
$Cosh^{-1}$	**One**	$F(Y)=\frac{2}{\exp(Y1)+\exp(-Y1)}$
$Tanh^{-1}$	**One**	$F(Y)=\frac{\exp(Y1)+\exp(-Y1)}{\exp(Y1)-\exp(-Y1)}$

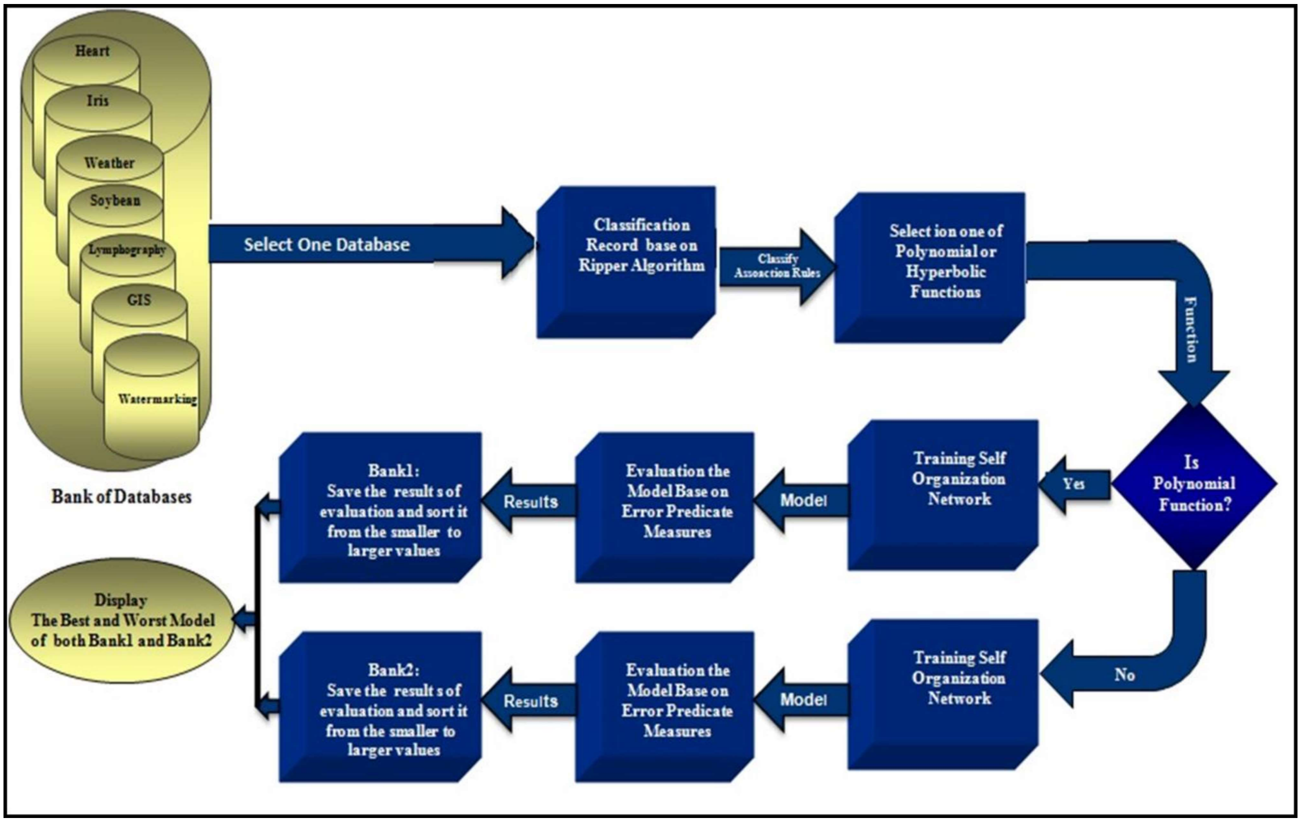


Fig 1: Block Diagram of the Meta Knowledge System

In general, there are two kinds of loops in the Ripper algorithm

- Outer loop : adding one rule at a time to the rule base
- Inner loop : adding one condition at a time to the current rule

Conditions are added to the rule to maximize an information gain measure. Conditions are added to the rule until it covers no negative example. In the this algorithm, conditions are added to the rule to Maximize an information gain measure

$$Gain(R', R) = s \cdot (\log_2 \frac{N'_+}{N'} - \log_2 \frac{N_+}{N})$$

Where, R : the original rule, R' : the candidate rule after adding a condition, N (N'): the number of instances that are covered by R (R'), N_+ (N'_+): the number of true positives in R (R'), s : the number of true positives in R and R' (after adding the condition),

Until it covers no negative example

$$rvm(R) = \frac{p-n}{p+n} \approx 1$$

Where, *Rvm is* Rule value metric. p and n : the number of true and false positives respectively. Table 2 shows the comparison of rule- based classifiers. This comparison is based on the following points (Rule Growing Strategy, Evaluation Metric, Stopping Condition for rule growing, Rule Pruning, Instances elimination, stopping condition for adding rules, Rule Set Pruning, and Search Strategy).

Table 2: Comparison Of Rule-Base Classifiers

	RIPPER	CN2 (unordered)	CN2 (ordered)	AQR
Rule-growing strategy	General-to-specific	General-to-specific	General-to-specific	General-to-specific (seeded by a positive example)
Evaluation Metric	FOIL's Info gain	Laplace	Entropy and likelihood ratio	Number of true positives
Stopping condition for rule-growing	All examples belong to the same class	No performance gain	No performance gain	Rules cover only positive class
Rule Pruning	Reduced error pruning	None	None	None
Instance Elimination	Positive and negative	Positive only	Positive only	Positive and negative
Stopping condition for adding rules	Error > 50% or based on MDL	No performance gain	No performance gain	All positive examples are covered
Rule Set Pruning	Replace or modify rules	Statistical tests	None	None
Search strategy	Greedy	Beam search	Beam search	Beam search

3. EXPERIMENTS

To test the performance of the suggested system, we use seven databases (i.e., weather, Iris, Heart, Soybean, Lymphography, GIS and Watermarking) which are different in natural, size, number of samples, number of features, and types of uses as explained in Table 3.

Table 3 : Description Of The Databases Used To Test MKS

Name of Database	Attribute Characteristics	Associated Tasks	Number of Instances	Number of Attribute	Area
Heart	Categorical	Classification	278	14	Heath
Iris	Integer, Real	Classification	121	5	Medical
Weather	Categorical	Classification	14	5	Weather
Soybean	Integer, Binary	Classification	27	35	Life
Lymphography	Integer	Classification	17	19	Medical
GIS	Integer, Real	Classification	1001	9	Geographic
Watermarking	Real, Categorical	Classification	3360	6	Image

The heart database is taken as an example to explain the behaviors of the suggested system. The following rules generation by RIPPER, whiles the surface of classification Heart database explained in Figure 2 (i.e., figure divided the state of pation into three classes “normal, fix and Rev”. based on the main three features inflect in taken the decision Heart rate, Cholesterol And Chestpain) .

<u>Set of Rules Generation by RIPPER</u>

- **Rule1:** If HEARTRATE <= 150.5 and CHOLESTEROL <= 246.5 and CHESTPAIN <= 0.665 and OLDPEAK <= 1.5 Then class = normal
- **Rule2:** If HEARTRATE <= 150.5 and CHOLESTEROL <= 246.5 and CHESTPAIN <= 0.665 and OLDPEAK > 1.5 and OLDPEAK <= 3.1 Then class = fix
- **Rule3:** If HEARTRATE <= 150.5 and CHOLESTEROL <= 246.5 and CHESTPAIN <= 0.665 and OLDPEAK > 3.1 Then class = Rev .
- **Rule4:** If HEARTRATE <= 150.5 and CHOLESTERAL <= 246.5 and CHESTPAIN > 0.665 Then class = fix
- **Rule5:** If HEARTRATE <= 150.5 and CHOLESTERAL > 246.5 and SLOPE <= 0.665 Then class = Rev
- **Rule6:** If HEARTRATE <= 150.5 and CHOLESTERAL > 246.5 and SLOPE > 0.665 Then class = fix
- **Rule6:** If HEARTRATE <= 150.5 and CHOLESTERAL > 246.5 and SLOPE > 0.665 Then class = fix
- **Rule7:** If HEARTRATE > 150.5 Then class = normal

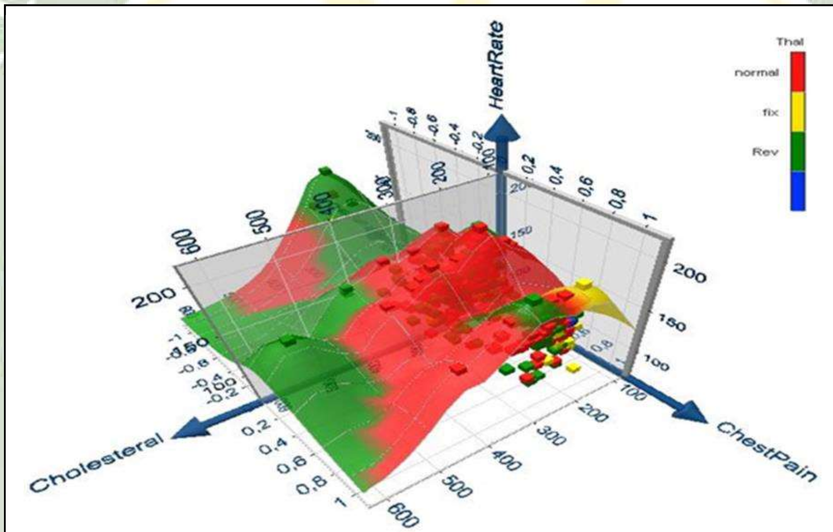


Figure 2: Surface of Heart Database

Table 4 shows the mathematical models generation by MKS by polynomial functions while Table 5 explains the analysis of the polynomial models based on five predicate error measures(i.e., Maximum error, RMSE, MSE, MAE, MAPE).

Figure 3 shows Analysis of Polynomial Models resulting from the MKS based on Predicating Error Models of Heart Database.

Table 6 explains the mathematical models by hyperbolic functions, while Table 7 explains the analysis of hyperbolic models based on five predicate error measures. Finally, Table 8 gives an analysis of all databases tested by the MKS system.

Table 4: Mathematical Models Generation By MKS Using Polynomial Functions

POLYNOMIAL FUNCTIONS	MATHEMATICAL MODELS
Linear: one variable	**Class = 2.5-0.5*CHOLESTEROL[126,246.5]**
Linear: two variables	**Class = -1.44329e-015-1.429787e+015*S(2)+1.429787e+015*S(4)** S(1) = 2.5-0.5*OLDPEAK[0,1.5]-3.330669e-016*CHESTPAINS[0.666,1] S(2) = -1.5-1*CHOLESTEROL[126,246.5]+2*S(1) S(3) = 3.5-1*OLDPEAK[0,1.5]-1*CHOLESTEROL[126,246.5] S(4) = 4.940492e-015-9.714451e-016*OLDPEAK[0,1.5]+1*S(3)
Linear: three variables	**Class = 1.007574-0.114719*SLOPE+1.054781*S(3)-0.064081*S(6)** S(1) = 0.275289-0.185214*SLOPE+0.000679*CHOLESTEROL-0.113501*CHESTPAINS S(2) = 1.080789+0.00896*OLDPEAK-0.004578*HeartRate-0.105058*CHESTPAINS S(3) = -0.238834+0.000566*OldPeak+0.668943*N(1)+0.937531*S(2) S(4) = 1.21335-0.216435*SLOPE-0.005176*HeartRate-0.104305*CHESTPAINS S(5) = 0.400905-0.152695*SLOPE+0.011891*OLDPEAK-0.129856*CHESTPAINS S(6) = -0.190552+0.06522*SLOPE+0.879729*S(4)+0.603062*S(5)
Quadratic: one variable	**Class = 2.5-94.70782*CHOLESTEROL[126,246.5]+ 94.20782*CHOLESTEROL[126 ,246.5]^2**
Quadratic: two variables	**Class = 1-7.910155e+017*CHESTPAINS[0.666,1]-2.806969e+018* CHESTPAINS[0.666, 1] ^2- 3.374474e+033*HEARTRATE[71,150.5]+3.374474e+033 *HEARTRATE [71,150.5]^2+3.142844e+018*CHESTPAIS[0.666,1]*HEARTRATE[71,150.5]**
Cubic: one variable	**Class = 2.2+2.322201e+015*SLOPE[0.666,1]+6.276303e+032*SLOPE[0.666,1]^2- 6.276303e+032*SLOPE[0.666,1]^3**
Product: two variables	**Class = -3.206456+1.15221*S(1)*S(3)** (1) = 2.2-0.2*CHESTPAINS[0.666,1]*HEARTRATE[71,150.5] S(2) = 2.498385-0.231432*CHOLESTEROL[126,246.5]*S(1) S(3) = 2.498385-0.231432*OLDPEAK[0,1.5]*S(2)
Logistic: one variable	**Class=1.680114+2.986964/(1+exp(1.148794*(CHOLESTEROL[126,246.5]+0.846093)))**
Log: one variable	**Class = 2.234401-0.504937*log(CHOLESTEROL[126,246.5]+0.591045)**
Exponential: one variable	**Class = -0.721551+0.401219*exp(0.828924*(S(1)-2.19709))** S(1) = 1.099382+1.400618*exp(-0.156504*(CHOLESTEROL[126,246.5]+0.1))
Asymptotic: one variable	**Class = 2.048245-0.043618/(CHOLESTEROL[126,246.5]-0.096489)**

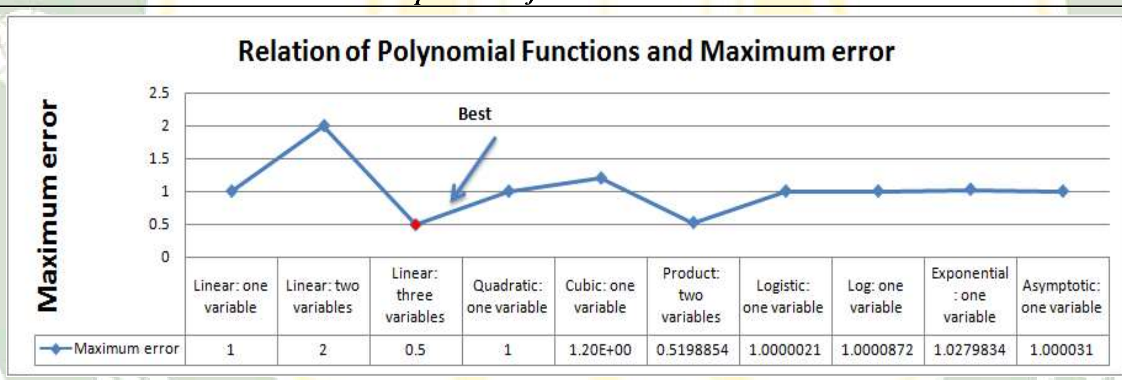
Table 5: Analysis of Polynomial Models Based On Five Predicating Error Measures

Polynomial Functions	Maximum error	RMSE	MSE	MAE	MAPE
Linear: one variable	1	0,7071068	0,5	0,5714286	39,2857140
Linear: two variables	2	1,0394023	1,0803571	0,75	33,9285710
Linear: three variables	**0,5**	**0,2672612**	**0,0714286**	**0,1428571**	**5,9523810**
Quadratic: one variable	1	0,7071068	0,5	0,5714286	39,2857140
Quadratic: Two variables	39	483	33	312	211
Cubic: one variable	1,20E+00	0,7819390	6,11E-01	0,6571429	46,1904760
Product: two variables	0,5198854	0,4531915	0,2053825	0,4483228	25,1713910
Logistic: one variable	1,0000021	0,7071064	0,4999994	0,5714289	39,2856700
Log: one variable	1,0000872	0,7070892	0,4999751	0,5714410	39,2847150
Exponential: one variable	1,0279834	0,7020728	0,4929062	0,5754262	38,9368200
Asymptotic: one variable	1,000031	0,7071005	0,4999912	0,5714330	39,2861110

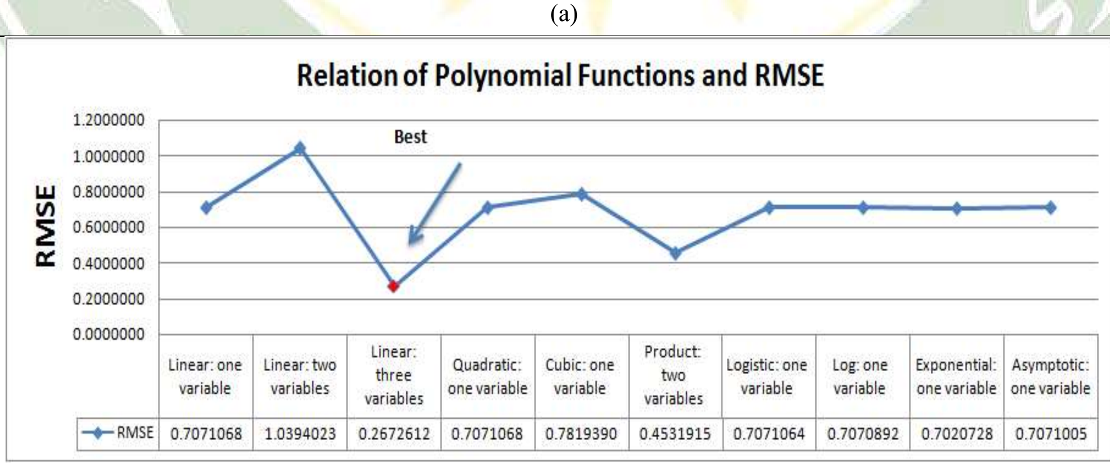
Best Model

Worst Model

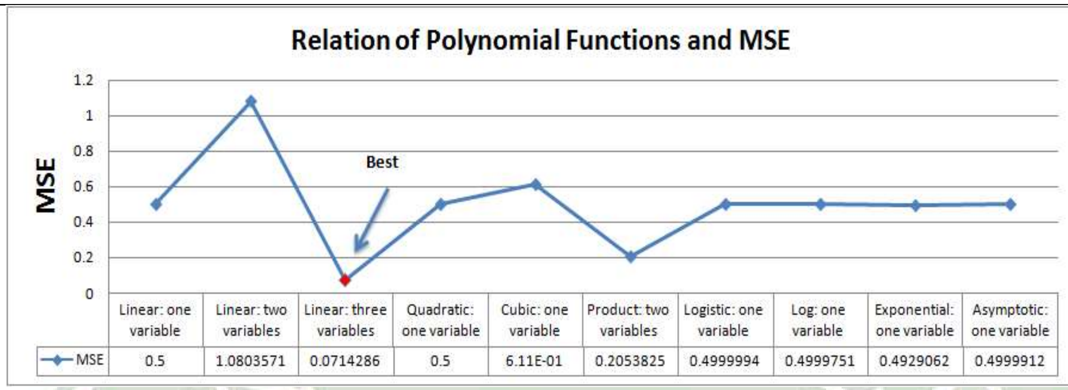
The results in Table (5) showed the ***best model*** of heart database using polynomial functions is ***linear of three variables*** while the ***worst model*** is ***<u>quadratic of two variables</u>***.



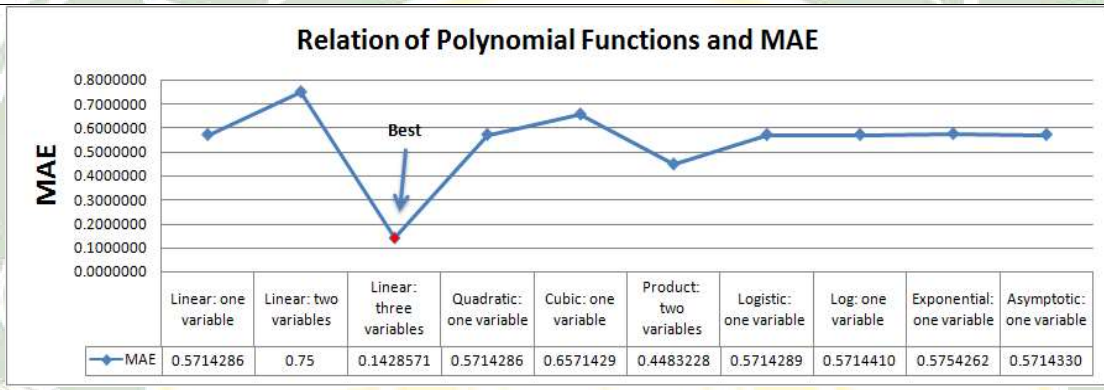
(a)



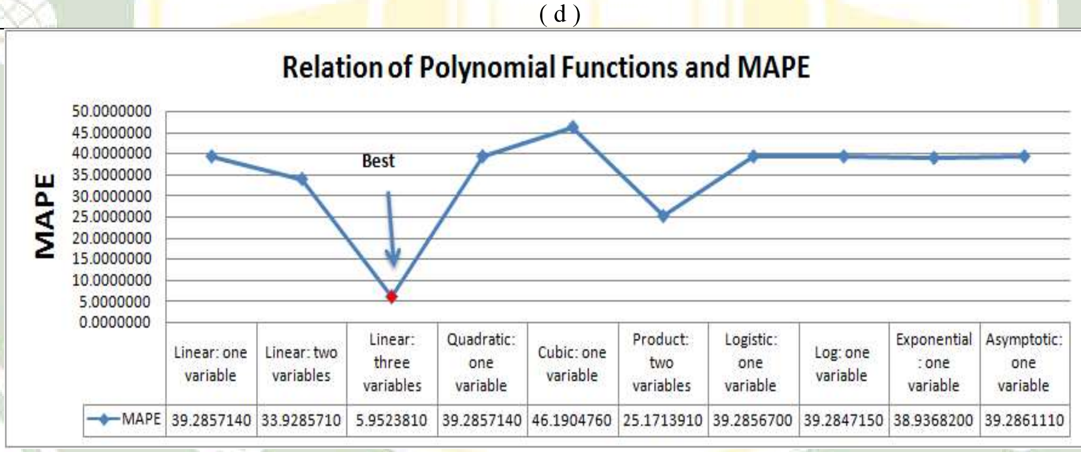
(b)



(c)



(d)



(e)

Fig 3: Analysis of Polynomial Models resulted from the MKS based on Predicating Error Models of Heart Database. (a) Relation of Polynomial Functions and Maximum error, (b) Relation of Polynomial Functions and RMSE, (c) Relation of Polynomial Functions and MSE, (d) Relation of Polynomial Functions and MAE and (e) Relation of Polynomial Functions and MAPE.

Table 6: Mathematical Models By Hyperbolic Functions

Hyperbolic Functions	MATHEMATICAL MODELS BASED ON HYPERBOLIC FUNCTIONS
Sinh: One Variable	Class =a0+a1* Sinh (4-HEARTRATE^0.2)+ a2* Sinh ((CHOLESTERAL* 0.825/ 0.5)) +a3*0.2* Sinh((CHESTPAIN^0.2)-0.7)+ a4* Sinh ((OLDPEAK* 0.6/ (0.382^0.5))+ a5* Sinh (14-SLOPE^0.5)
Cosh: One Variable	Class =a0+a1* Cosh (HEARTRATE-0.4493^0.2)+ a2* Cosh (-(CHOLESTERAL* ^2)+0.734) +a3* Cosh (CHESTPAIN*0.5)+ a4* Cosh ((OLDPEAK*0.9/ (0.3^0.5)) + a5* Cosh (SLOPE+0.678)
Tanh: One Variable	Class =a0+a1* 0.8*Tanh (HEARTRATE*0.2)+ a2* Tanh (CHOLESTERAL+22.4) +a3* 4/(Tanh (CHESTPAIN*0.5))+ a4* Tanh ((OLDPEAK*0.5+1)) + a5* Tanh (SLOPE^2)
Sinh-1: One Variable	Class =a0+a1* Sinh-1(0. 4/(HEARTRATE -0.09)+ a2* Sinh-1(0.82*(CHOLESTERAL -2.19)+ a3*Sinh-1(-0.216435* CHESTPAIN)+ a4*Sinh-1(-((OLDPEAK *0.005176)^2)+ a5*Sinh-1 (-(0.104305* SLOPE))
Cosh-1: One Variable	Class =a0+a1* Cosh-1 (0. 982* HEARTRATE)+ a2*0.6* Cosh-1(CHOLESTERAL +14.5)+ a3* Cosh-1 (-(0.5* CHESTPAIN))+ a4* Cosh-1 (OLDPEAK +14.5*0. 6)+ a5* Cosh-1 ((SLOPE+0.372)/0.281)
Tanh-1: One Variable	Class =a0+a1*0,144* Tanh-1 (HEARTRATE)+ (a2*0.6)/ Tanh-1 (CHOLESTERAL - 0.185^2) + a3* Tanh-1 (CHESTPAIN+0.401219) + a4* Tanh-1 (OLDPEAK +0.048826)+ a5* Tanh-1 (0.32*SLOPE^3)

Table 7: Analysis Of Hyperbolic Models Based On Five Predicate Error Measures

Hyperbolic Functions	Maximum error	RMSE	MSE	MAE	MAPE
Sinh: One Variable	1.4401730	1.1982734	0.6901638	0.9087240	42.68401
Cosh: One Variable	1.5283001	1.2037485	0.7635219	0.9267195	45.82734
Tanh: One Variable	1.4922187	1.0763549	0.7183540	0.7562813	43.90928
$Sinh^{-1}$: One Variable	1.3723642	1.1445210	0.6700913	0.8273588	45.00836
$Cosh^{-1}$: One Variable	1.3889263	0.9834652	0.5637251	0.8920054	44.54719
$Tanh^{-1}$: One Variable	**1.0000586**	**0.7070950**	**0.4999833**	**0.5714369**	**39.28497**

Worst Model (Cosh: One Variable)

Best Model ($Tanh^{-1}$: One Variable)

The results in Table (7) showed the ***best model*** of that Heart database by using hyperbolic functions is ***Inver of Tanh of one variable*** while the ***worst model*** is ***Cosh of one variable***.

Table 8 :Analysis Of All Models Tested By MKS System

Name of Database	Number of classified rules	Number of Attributes	Polynomial Models		Hyperbolic Model	
			Best	Worst	Best	Worst
Heart [16]	7	5	Linear: three variables	Quadratic : two variables	Inver of Tanh: one variable	Cosh: one variable
Iris [17]	5	2	Linear: two variables	Cubic: one variable	Inver of Tanh: one variable	Sinh: one variable
Weather	4	3	Quadratic: two variables	Cubic: one variable	Inver of Tanh: one variable	Cosh: one variable
Soybean [18]	7	6	Quadratic: two variables	Cubic: one variable	Inver of Tanh: one variable	Sinh: one variable
Lymphography [20]	12	10	Linear: three variables	Cubic: one variable	Inver of Tanh: one variable	Sinh: one variable
GIS [21]	25	5	Linear: three variables	Quadratic : one variable	Tanh: one variable	Inver of Sinh: one variable
Watermarking [19]	15	4	Linear: three variables	Quadratic : one variable	Tanh: one variable	Inver of Tanh: one variable

4. DISCUSSION

In this section, we attempt to answer about some of equations relate of that work:

Why need to generate mathematical models? We need of it, because the mathematical models are easy prove, simplification and combination.

Is RIPPER suitable of that work and why? RIPPER stands for Repeated Incremental Pruning to Produce Error Reduction. This algorithm. It is based on association rules with reduced error pruning (REP), a very common and effective technique found in decision tree algorithms. In REP for rules algorithms, the training data is split into a growing set and a pruning set. First, an initial rule set is formed that is the growing set, using some heuristic method. This overlarge rule set is then repeatedly simplified by applying one of a set of pruning operators typical pruning operators would be to delete any single condition or any single rule. At each stage of simplification, the pruning operator chosen is the one that yields the greatest reduction of error on the pruning set. Simplification ends when applying any pruning operator would increase error on the pruning set [10]. Therefore; RIPPER very suitable to generation the classified rules.

Why this paper focused on develop self-organization network? Because, we found the connections between the neurons in the network are not fixed but rather are selected during training to optimize the network. The number of layers in the network also is selected automatically to produce maximum accuracy without over fitting.

Can we use other evaluation measures to test the accuracy rate of models? Yes, we suggest experimenting on the new measures to select the best mathematical models from the polynomial or hyperbolic family of models. In general, our advice is to use the three predicting measures of coefficient of multiple determination (R^2P), residual mean square (MSep) and Malo statistical value CP to differentiate among the product models to obtain more precise or better predicting outcomes.

A. The coefficient of multiple determination(R^2P)

$$R^2P = \frac{SSR(X1, X2, \ldots\ldots, Xp)}{SST} = 1 - \frac{SSe(X1, X2, \ldots\ldots, Xp)}{SST}$$

B. Residual mean square (MSep)

$$MScp = \frac{SSe(X1, X2, \ldots\ldots, Xp)}{n - p}$$

C. Malo statistical value CP

$$CP = \frac{SSe(X1, X2, \ldots\ldots, Xp)}{MSR(X1, X2, \ldots\ldots, Xm)} - (n - 2p)$$

5. CONCLUSION

This work combined between the advantages of data mining algorithms and self-organizing neural network. In the design mathematical models phase, the proposed method can be considered as a meta- knowledge method, this was extracting new knowledge (i.e., mathematical models) from the original knowledge (i.e., classified rules).

The main benefit of MKS is given the user' vision about the best model can be used based on the name of database. Therefore, in the future the users not need working on the all the different models. In addition, reduce the time of implementation.

The paper from experiments can provide any reader by ***important two reports*** include:

First; all the huge databases(i.e., DBs use in experiments) have the fix behavior, the best model of their generated by linear of three variables function and worst model of their generated by cubic of one variable or quadratic function related to polynomial models. Add to that, the best model of their generated by tanh of one variable function and worst model of their generated by more than of one other functions related to hyperbolic models.

Second; all the small databases(i.e., DBs use in experiments) have the not stabile behavior, the best model of their generated by linear of three and two variables or quadratic of two variables function and the worst model of their generated by cubic of one variable or quadratic function related to polynomial models. Add to that, the best model of their generated by inverse of tanh of one variable function and the worst model of their generated by sinh or cosh functions related to hyperbolic models.

Of course, all the above conclusions establish base on compute the five of the main predicate error measures of the design mathematical models. These predicate measures (Maximum error, RMSE, MSE, MAE, MAPE).

As a result, we found that by combining the design dynamic mathematical models led to an increase of the accuracy of results, but, on other hand require to have good background of mathematical principles.

REFERENCES

[1] G.Nagarajan, K.K. Thyagharajan, " A Machine Learning Technique for Semantic Search Engine", Proccedia Engineering 38 ,ELSEVIER, **2012**.

[2] StatSoft electronic statistics textbook, 2010 , http://www.statsoft.com/textbook .

[3] Anand R., Jure L. and Jeffrey D. “Mining of Massive Datasets”, **2011**, http://infolab.stanford.edu/~ullman/mining/mining.html

[4] Rada Mihalcea, Andras Csomai, " Wikify! Linking Documents to Encyclopedic Knowledge ", **2007 .**

[5] Rupal Gupta , Sanjay Kumar Malik, " SPARQL Semantics And Execution Analysis In Semantic Web Using Various Tools " **, 2011**

[6] Barak Chizi and Oded Maimon,” Dimension Reduction and Feature Selection”, Data Mining and Knowledge Discovery Handbook, 2nd ed., Springer Science and Business Media, LLC **2010**.

[7] Vitaly Schetinin, Polynomial Neural Networks Learnt to Classify EEG Signals, NIMIA-SC2001 - NATO Advanced Study Institute on Neural Networks**, 2001**

[8] André Freitas,Edward Curry,João Gabriel Oliveira,and Seán O’Riain, "Querying Heterogeneous Datasets on the Linked Data Web Challenges, Approaches, and Trends", **2012**, IEEE.

[9] Tetyana. Aksyonova, Vladimir V. Volkovich And Igor V. Tetko” Robust Polynomial Neural Networks In Quantative-Structure Activity Relationship Studies, Systems Analysis Modelling Simulation, Vol. 43, No. 10, October **2003**, pp. 1331-1339

[10] Seong-Hwan Jun , Boosted regression trees and random forests , Statistical Consulting Report , **2013** .

[11] Cohen, W., “Fast effective rule induction”. Proceedings of International Conference on machine Learning 1995, pp.1-10.

[12] Harrington and Peter, Book, “Machine Learning In Action", Manning publishing, **2012.**

[13] Alex Smola, S.V.N. Vishwanathan, " Introduction to Machine Learning", Cambridge University Press, **2008.**

[14] Jing Ding , Shanlin Yang , Classification Rules Mining Model with Genetic Algorithm in Cloud Computing , International Journal of Computer Applications Volume 48– No.18 , **2012.**

[15] Kawuu W. Lin , Yu-Chin Lo , Efficient Algorithms For Frequent Pattern Mining In Many-Task Computing Environments , ELSEVEIR Knowledge-Based Systems , **2013.**

[16] Diagnoses the medical cases, the LUPEM hospital, UK ,**2005.**

[17] http://sites.google.com/site/yaoxuchen/Home/ iris .**2010.**

[18] Center for Machine Learning and Intelligent Systems .USA. http://archive.ics.uci.edu/ml/machine_learningdatabases/ soybean/**1998**.

[19] Laboratoire d'information de Robotique et de Microelectronique d Montpellier(LIRMM), France,**2011.**

[20] Michalski,R.S. Learning Doug Fisher (dfisher%vuse@uunet.uucp),**2003**.

[21] Center for Machine Learning and Intelligent Systems .USA. http:// ://idke.ruc.edu /GIS/**2010**