Asian Journal of Scientific Research

ISSN 1992-1454

science alert
http://www.scialert.net

ANSInet
an open access publisher
http://ansinet.com

Asian Journal of Scientific Research 5 (3): 121-132, 2012
ISSN 1992-1454 / DOI: 10.3923/ajsr.2012.121.132

A Rule Extraction Algorithm That Scales Between Fidelity and Comprehensibility

[1]Kalaiarasi Sonai Muthu Anbananthen, [3]Fabian Chan Huan Pheng, [2]Subhacini Subramaniam, [1]Shohel Sayeed and [1]Eimad Eldin Abdu Ali Abusham
[1]Faculty of Information Science and Technology (FIST), [2]Centre for Diploma Programme, Multimedia University, Melaka, Malaysia
[3]INTI College Sabah, Sabah, Malaysia

Corresponding Author: Kalaiarasi Sonai Muthu, Faculty of Information Science and Technology (FIST), Multimedia University, Melaka, Malaysia

ABSTRACT

Fidelity and comprehensibility are the common measures used in the evaluation of rules extracted from neural networks. However, these two measures are found to be inverse relations of one another. Since the needs of comprehensibility or fidelity may vary depending on the user or application, this paper presented a significance based rule extraction algorithm that allows a user set parameter to scale between the desired degree of fidelity and comprehensibility of the rules extracted. A detailed explanation and example application of this algorithm is presented as well as experimental results on several neural network ensembles.

Key words: Comprehensibility, fidelity, rule extraction, ensemble networks

INTRODUCTION

Artificial Neural Network (ANN) has been applied in many applications with remarkable success (Shahrabi *et al*., 2009; Khatib and Al-Sadi, 2011; Shakiba *et al*., 2008; Tanoh *et al*., 2008). For example, ANN have been successfully applied in the area of prediction (Senol and Ozturan, 2010), handwritten character recognition (Lotfi and Benyettou, 2011; Khanale and Chitnis, 2011), evaluating prices of housing (Eriki and Udegbunam, 2010), disease classification (Tahir and Manap, 2012). Although most of the results that can be achieved through the application of neural networks are remarkable and have frequently found to outperform traditional approaches but the main limitation of artificial neural network is not transparent.

The needs and potential benefits of providing transparency to artificial neural networks and more recently ensembles have been expounded upon time and again in past literature (Andrews *et al*., 1995; Tickle *et al*., 1997; Wall *et al*., 2002). Among the measures used to evaluate rule extraction algorithms are the fidelity and comprehensibility of the rules extracted. As defined by Andrews *et al*. (1995), fidelity refers to how well the rule set is able to mimic the behavior of the neural network in terms of its classification whereas comprehensibility is measured based on the number of rules and the number of antecedents per rule. However, researchers of rule extraction techniques have since discovered that these two measures are often in conflict (Tickle *et al*., 1997). The emphasis on either measure typically caused a decline in the other, particularly in

decompositional rule extraction approaches. It is common therefore that some rule extraction algorithms sacrifice fidelity in return for better rule comprehensibility as seen in the termination of third order rules in the rule extraction algorithm presented by (Tsukimoto, 2000).

In practice however, the desired level of fidelity and comprehensibility can vary according to the needs of the user or the particular application for which the rule extraction is being employed. For example, safety critical systems will probably put more emphasis on fidelity in order to obtain as accurate as possible a representation of the model. A maintenance prediction system on the other hand may only require a few general rules to decide the most urgent parts that need replacement or maintenance. The inverse relationship of these two measures makes it possible to scale between them according to the needs of the user.

This study therefore proposed a rule extraction algorithm based on a significance measure calculated from weight links which allows a scaling between the two criterions of fidelity and comprehensibility. The rule set extracted from this algorithm is viewed as an approximation to the neural network of which the degree of approximation can be adjusted via a user set parameter. A closer approximation gives better fidelity but poorer comprehensibility and likewise a more general approximation gives better comprehensibility but poorer fidelity. This allows the user to decide how detailed or general a rule set is desired.

RULE EXTRACTION APPROACH

The rule extraction approach used in this paper requires some modification in the measures or criteria presented by Andrews *et al.* (1995) and Tickle *et al.* (1997). This is mainly due to the fact that the rule extraction algorithm here is also targeted at ensemble neural networks. Applying rule extraction to ensembles presents new problems not faced in single neural networks. One particular aspect is diversity. The increase of accuracy and generalization capability of ensembles is attributed to the diversity of its components. The combination of several individual network components in an ensemble allows the diversity of the components to overcome individual component errors. This also means that each ensemble output may utilize the prediction capability of different components. The typical rule extraction approaches used in the past attempts to extract a holistic rule set which can ideally perform as a surrogate for the neural network. Due to the fact that each instance of a classification in an ensemble may use different components the rule extraction approach here uses a case by case basis. Although a holistic rule extraction approach would produce a much better rule set by which we can study the model, this may not be necessary in applications where real time transparency is needed on a case by case basis.. Another author using this approach to rule extraction for ensembles is given by Wall and Cunningham (2000). This approach affects the measure of rule classification accuracy in that since the rules are extracted for each instance of data passed into the ensemble, the rule set produced is in fact specific to that data and not intended to be holistic. The rule classification accuracy used in this paper therefore measures the accuracy of the rule set produced against the targets of the data. Another measure which is not much affected but modified in this approach is comprehensibility.

Although, comprehensibility is still evaluated in terms of rules and antecedents, since the rule extraction algorithm will extract a rule for each instance of data, the theoretical maximum number of antecedents in the final rule set can be computed as the number of instances times the number of attributes of the data. In a binary classification dataset this is further reduced as we only need to extract a rule for each positive classification since unclassified instances can by default be set as negative. The extracted rule sets are thus measured as a percentage of this theoretical maximum.

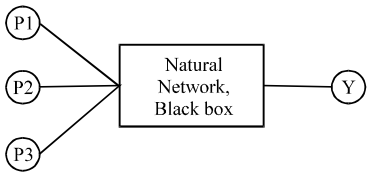

Fig. 1: A typical neural network

Due to this measure of comprehensibility, understandability of the rules and comprehensibility are not necessarily synonymous. Consider a typical neural network shown in Fig. 1 where P1, P2 and P3 are the network inputs and Y is the network output.

Suppose that the neural network is fed the input of P1 = 1, P2 = 2 and P3= 3 and produces an output of Y = 1. Using a rule extraction algorithm on the neural network will then produces the following rule result.

IF P1 = 1 and P2 = 2 and P3 = 3 then Y = 1

The single line of rule having only 3 antecedents is easily understandable but according to the measure of comprehensibility defined earlier would represent poor comprehensibility. A rule that represents all the inputs to the network does not give added comprehensibility to the model because that information is apparent without the use of any rule extraction algorithm.

This modified measure of comprehensibility allows a better evaluation of the fidelity to comprehensibility relationship which aids in the adjusting of the approximation parameter.

RULE EXTRACTION ALGORITHM

This rule extraction algorithm makes use of a significance measure calculated for each weight link in the network by layer and repeatedly aggregating them with each preceding layer until the significance of the inputs being fed into the network can be determined. The basis for the calculation of this significance value is that each neuron receives inputs along several links but some links are more significant than other links. This significance is determined by their input weight product value divided by the total sum of input weight products for that particular neuron. A modulus operation is used on the input weight products because we are only interested in comparing the absolute value of each link. The significance value for any weight link can thus be expressed in Eq. 1.

$$\text{Significance of } w_i p_i = \frac{|w_i p_i|}{\sum_{i=1}^{n} |w_i p_i|} \quad (1)$$

where, $w_i p_i$ are respectively the ith weights and inputs of a neuron from 1 to n number of inputs.

The aggregation of significance between the layers of the neurons is accomplished by culminating the significance from the output layer back to the input layer following the respective links of the neurons. Assuming a two layer network the significance equation can be expressed as given in Eq. 2 although this equation can be expanded to any number of layers depending on the network:

$$\text{Culminated significance of } w_i p_i = \frac{|w_i p_i|}{\sum_{i=1}^{n} |w_i p_i|} \times \frac{|v_j h_j|}{\sum_{j=1}^{n} |v_j h_j|} \quad (2)$$

where, $w_i p_i$ are respectively the ith weights and inputs to a hidden neuron from 1 to n number of inputs and $v_j h_j$ are respectively the jth weights and inputs to an output neuron from 1 to n number of hidden neurons.

From Eq. 1 and 2 it can be seen that the sum of the significance or culminated significance for any layer of links always equals 1. When the significance has been aggregated back to the input layer the significance of the inputs can be expressed as given in Eq. 3:

$$\text{Significance of } p_i = \sum_{j=1}^{n} CS(w_{ij} p_i) \quad (3)$$

where, p_i is the ith input and CS stands for the Culminated Significance Eq. 2 and $w_{ij} p_i$ are respectively the weights and input from the ith input to the jth hidden neuron from 1 to n number of hidden neurons.

Since the equation used to determine the significance essentially uses a ratio formula, the significance values therefore has a range from 0 to 1. A user set threshold within a range of 0 to 1 is used to determine the minimum acceptable significance value for an input to be considered significant to the network output. The significance based rule extraction algorithm is given in Table 1.

This section shows a worked out example of the rule extraction algorithm applied on a neural network trained on the buy stock data set which is shown in Table 1. The buy stock data set is a mock data set used because its small size makes it feasible for calculations by hand and manual evaluation of the soundness of an algorithm. The inputs of the buy stock data set shown in Table 1 have already been normalized to fall between the values of -1 and 1 as the calculations later will use these normalized values.

A neural network with the architecture shown in Fig. 2 was trained on the buy stock dataset given in Table 2. All weight links and biases are given and the activation functions used at the hidden and output layers are log sigmoid.

Input to hidden layer weights (w_{ij}):

$$w_{11} = 1.0354 w_{12} = 3.4246 W_{13} = -4.5933$$
$$w_{21} = -0.0975 w_{22} = -0.7046 w_{23} = -0.1997$$
$$w_{31} = 4.5009 w_{32} = 0.4394 w_{33} = -2.1320$$
$$w_{41} = 0.4157 w_{42} = 2.0849 w_{43} = -0.2304$$

Table 1: Significance based rule extraction algorithm

1	Calculate the significance of the links in the last layer of the network
2	Calculate the culminated significance of the links in the preceding layer
3	Repeat step 2 until the culminated significance of the links in first layer of the network is obtained
4	Calculate significance of the inputs
5	Determine significant inputs for rule extraction based on user set threshold
6	Output results in a binary input usage vector

Table 2: Buy stock dataset

Outlook	Price	Market	Earning	Buy?
-1	-1	-1	-1	1
-1	-1	-1	1	1
0	-1	-1	-1	0
1	0	-1	-1	0
1	0	1	-1	0
1	0	1	1	1
0	0	1	1	0
-1	1	-1	-1	1
-1	0	1	-1	0
1	1	1	-1	0
-1	1	1	1	0
0	1	-1	1	0
0	-1	1	-1	0
1	1	-1	1	1

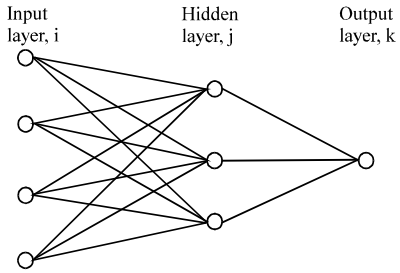

Fig. 2: Buy stock neural network architecture

Hidden to output layer weights (v_{jk}):

$$v_{11} = -.6768 v_{21} = 5.9628 v_{31} = 5.6798$$

Hidden layer biases (b_j):

$$b_1 = -2.274 b_2 = -3.1910 b_3 = -4.9482$$

Output layer bias (c_k):

$$c_1 = -2.4733$$

The rule extraction algorithm extracts rules for each positive output of the network. The simplicity of this dataset allows the neural network to achieve full accuracy and thus the outputs of the network are identical to the targets in the buy stock data set. Starting with the first positive output of the network, we have the following inputs (p_i) being fed into the network.

- $p_1 = -1$ $p_2 = -1$ $p_3 = -1$ $p_4 = -1$

The algorithm begins by calculating the hidden neuron outputs (h_j):

$$h_j = \text{Logsig}\left(\sum_{j=1}^{n} w_{ij}p_1 + b_j\right)$$

$h_1 = \text{Logsig}\ (w_{11}p_1+w_{21}p_2+w_{31}p_3+w_{41}p_4+b_1) = \text{Logsig}\ ((1.0354\times\text{-}1)+(\text{-}0.0975\times\text{-}1)+(4.5009\times\text{-}1)+(0.4157\times\text{-}1)+(\text{-}2.2747) = \text{Logsig}\ (\text{-}8.1292) = 0.0003$

$h_2 = 0.0002$

$h_3 = 0.9009$

The hidden neuron outputs are then used to calculate the significance of the weight links from the hidden to the output layer using (1):

$$\text{Significance of } v_{11}h_1 = \frac{|v_{11}h_1|}{|v_{11}h_1| + |v_{21}h_2| + |v_{31}h_{12}|} = \frac{|-1.6768\times 0.0003|}{|-1.6768\times 0.0003| + |5.9628\times 0.0002| + |5.6798\times 0.9009|} = 0.0001$$

Significance of $v_{21}h_2 = 0.0002$

Significance of $v_{31}h_3 = 0.9997$

Next the algorithm calculates the significance of weight links from the input to the hidden layer still using (1):

$$\text{Significance of } w_{11}p_1 = \frac{|w_{11}p_1|}{|w_{11}p_1| + |w_{21}p_2| + |w_{31}p_2| + |w_{41}p_2|} = \frac{|1.0354\times -1|}{|1.0354\times -1| + |-0.0975\times -1| + |0.41527\times -1|} = 0.1712$$

Significance of $w_{21}p_2 = 0.0161$
Significance of $w_{31}p_3 = 0.7440$
Significance of $w_{41}p_4 = 0.0687$
Significance of $w_{12}p_1 = 0.5147$
Significance of $w_{22}p_2 = 0.1059$
Significance of $w_{32}p_3 = 0.0660$
Significance of $w_{42}p_4 = 0.3133$
Significance of $w_{13}p_1 = 0.6419$
Significance of $w_{23}p_2 = 0.0279$
Significance of $w_{33}p_3 = 0.2980$
Significance of $w_{34}p_4 = 0.0322$

Having calculated the significance of links in both the input to hidden layer and hidden to output layers (2) can now be applied to get the culminated significance of the input to hidden layer weight links.

$$\text{Culminated significance of } w_{11}p_1 = \text{Significance of } w_{11}p_1 \times \text{Significance of } v_{11}h_1$$

$$= 0.1712 \times 0.0001$$

$$= 1.712 \times 10^{-5}$$

Culminated significance of $w_{21}p_2 = 1.61 \times 10^{-6}$
Culminated significance of $w_{31}p_3 = 7.44 \times 10^{-5}$
Culminated significance of $w_{41}p_4 = 6.87 \times 10^{-6}$
Culminated significance of $w_{12}p_1 = 1.029 \times 10^{-4}$
Culminated significance of $w_{22}p_2 = 2.118 \times 10^{-5}$
Culminated significance of $w_{32}p_3 = 1.32 \times 10^{-5}$
Culminated significance of $w_{42}p_4 = 6.266 \times 10^{-5}$
Culminated significance of $w_{13}p_1 = 0.6417$
Culminated significance of $w_{23}p_2 = 0.0279$
Culminated significance of $w_{33}p_3 = 0.2979$
Culminated significance of $w_{43}p_4 = 0.0322$

With no more preceding layers to aggregate, the algorithm uses the culminated significance of the input to hidden layer links to calculate a significance value for each input using (3):

$$\text{Significance of } p_1 = CS(w_{11}p_1)+CS(w_{12}p_1)+CS(w_{13}p_1)$$

$$= (1.712 \times 10^{-})+(1.029 \times 10^{-4})+0.6417$$

$$= 0.6418$$

Significance of $p_2 = 0.0279$
Significance of $p_3 = 0.2979$
Significance of $p_4 = 0.0323$

Finally the algorithm outputs a binary input usage using a user set threshold to decide which inputs are sufficiently significant:

$$\text{Input is significant if significance of } p_i > \text{threshold, Using a threshold of } 0.1$$

where, significance of $(p_1\ p_2\ p_3\ p_4) = (0.6418\ 0.0279\ 0.2979\ 0.0323)$:

$$\text{Binary input usage} = (p_1\ p_2\ p_3\ p_4)$$

$$= (1\ 0\ 1\ 0)$$

The main reason the algorithm uses a binary input usage is to enable rule combination in the ensemble voting process but an added advantage is that it makes it easy to store the rule set in a matrix form. This matrix form allows easy conversion of the rule set to the original data before normalization. Similarly using the original buy stock data set to generate the rule for this binary input usage the following rule is obtained:

IF outlook = 1 and market = 1 then buy = 1

Since the rule extraction algorithm runs on a case by case basis, the entire rule set is obtained only after repeating the rule extraction process on each instance of inputs to the network which produces a positive output. The subsequent rule set obtained using a threshold of 0.1 is shown in Fig. 3.

From the rule set shown in Fig. 3 it is obvious to see that the rule set can be summarized into just three rules by removing the redundant rules as shown in Fig. 4.

The effect of increasing the threshold used in the rule extraction algorithm can be seen in the summarized buy stock rule set in Fig. 5. The threshold used is 0.2 and the rule set contains less antecedents.

The rule sets extracted so far have 100% fidelity with the network classification which signifies that the rule sets have been able to classify the dataset with 100% accuracy. The cost of increasing comprehensibility however will cause a reduction in fidelity once a certain threshold value is breached which in the case of this dataset happens when we use a threshold value of 0.3. The summarized rule set is shown in Fig. 6 and has fidelity of 65%.

Since fidelity is reduced too much at a threshold of 0.3 we try the rule extraction algorithm with a threshold of 0.25. The summarized rules obtained are shown in Fig. 7 and the fidelity for the rules is again 100%.

It is interesting to note that the rule set obtained in Fig. 7 are the same positive traversals for the decision tree result of the ID3 algorithm shown in Fig. 8. This does not imply however that the rule extraction algorithm functions in the same manner as the ID3 algorithm but merely reinforces the validity of the rules obtained through this rule extraction algorithm.

IF outlook = 1 AND market = 1 THEN buy = 1
IF outlook = 1 AND market = 1 THEN buy = 1
IF outlook = 3 AND market = 2 AND earning = 2 THEN buy = 1
IF outlook = 1 AND market = 1 THEN buy = 1
IF outlook = 3 AND price = 3 AND earning = 2 THEN buy = 1

Fig. 3: Rules extracted at 0.1 threshold

IF outlook = 1 AND market = 1 THEN buy = 1
IF outlook = 3 AND market = 2 AND earning = 2 THEN buy = 1
IF outlook = 3 AND price = 3 AND earning = 2 THEN buy = 1

Fig. 4: Rules extracted at 0.1 threshold

IF outlook = 1 AND market = 1 THEN buy = 1
IF outlook = 3 AND market = 2 AND earning = 2 THEN buy = 1
IF outlook = 3 AND earning = 2 THEN buy = 1

Fig. 5: Rules extracted at 0.2 threshold

IF outlook = 1 THEN buy = 1
IF outlook = 3 THEN buy = 1
IF outlook = 3 AND earning=2 THEN buy = 1

Fig. 6: Rules extracted at 0.3 threshold

IF outlook = 1 AND market = 1 THEN buy = 1
IF outlook = 3 AND earning = 2 THEN buy = 1

Fig. 7: Rules extracted at 0.25 threshold

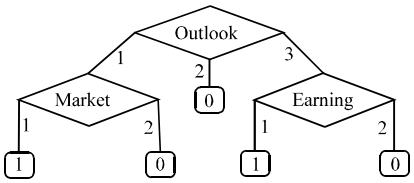

Fig. 8: ID3 output for buy stock dataset

EXPERIMENTATION

This section gives the experimentation results of the rule extraction algorithm applied on ensembles trained on 3 different datasets. The results of two different rule extraction algorithms for ensembles are compared.

Datasets: The datasets were obtained from UCI machine learning repository (2). Table 3 details the datasets used in the experiment.

Table 3: Dataset details

Dataset	Pima Indians diabetes	Breast cancer	Credit screening
No of instances	768	286	690
Used instances	768	277	653
Class Distribution	0 = 500	1 = 268	0 = 196
	1 = 81	0 = 357	1 = 296
Attributes	8	9	15
Division for train, test and validation set (%)	Train = 60	Train = 70	Train = 60
	Test = 20	Test = 20	Test = 20
	Val = 20	Val = 10	Val = 20

The used instances for Breast Cancer and Credit Screening are less than the number of instances available because all instances with missing values were removed. Attribute type for the datasets are continuous, discrete and a mix of continuous and discrete attributes respectively for Pima Indians Diabetes, Breast Cancer and Credit Screening.

Ensemble training: The ensembles were trained by generating a pool of 100 neural network components of which 5 of them are selected by genetic algorithm to constitute the ensemble. The ensemble training process is repeated 5 times for each dataset thus creating 5 unique ensembles for each dataset.

RESULTS

Rules were extracted from the ensembles trained on the datasets using two different rule extraction algorithms. The first algorithm identifies contributing inputs to a neuron by determining a minimum subset of links needed for a neuron to overcome its threshold (7). The subset of links determined is bounded by a 2 link limit to prevent too many links being removed. Table 4 shows the results using this algorithm.

The first column in Table 4 contains the names for the ensembles according to their dataset name and is numbered for each separate ensemble created for a particular dataset. Pidia, Bcan and Crscr each stand for Pima Indians Diabetes, Breast Cancer and Credit Screening dataset, respectively. EnsAcc is short for ensemble accuracy and shows the classification accuracy of the ensemble on the overall dataset. Similarly, Rule Acc is short for rule accuracy and shows the classification accuracy of the rule set on the dataset. Fid is short for fidelity and represents the accuracy of the rule classification output against the ensemble output. Rule No simply shows the number of rules in the rule set and Ante No shows the number of antecedents in the rule set. Note that the maximum number of antecedents possible for a rule set is equivalent to the number of rules in the dataset times the number of attributes in the dataset.

Table 5 shows the experiment results using the significance based algorithm. The ensembles from which the rules were extracted are identical to the ensembles used by the previous algorithm

Table 4: Results using minimum link subset algorithm with a 2 link limit

	Ens Acc	Rule Acc	Fid	Rule No.	Ante No.
Pidia1	80.60	80.60	100	217	1583
Pidia2	80.08	80.08	100	205	1625
Pidia3	80.08	80.08	100	223	1783
Pidia4	79.95	79.95	100	194	1529
Pidia5	79.82	79.82	100	267	2132
Bcan1	82.31	81.95	99.6	44	377
Bcan2	81.59	81.59	100	46	404
Bcan3	80.87	80.14	97.1	30	249
Bcan4	79.78	79.78	100	31	277
Bcan5	81.23	81.23	100	51	451
Crscr1	89.43	89.43	100	319	4699
Crscr2	88.36	88.36	100	306	4540
Crscr3	89.28	89.28	100	314	4598
Crscr4	88.82	88.82	100	315	4707
Crscr5	89.28	89.28	100	316	4709

Table 5: Results using significance based algorithm

	Ens Acc	Rule Acc	Fid	Rule No.	Ante No.
Pidia1	80.60	80.73	99.87	217	1063
Pidia2	80.08	80.21	99.87	205	983
Pidia3	80.08	80.08	100	223	1233
Pidia4	79.95	80.08	99.87	194	969
Pidia5	79.82	79.82	100	267	1296
Bcan1	82.31	77.62	93.86	44	264
Bcan2	81.59	77.26	94.22	46	276
Bcan3	80.87	76.53	93.50	30	178
Bcan4	79.78	76.17	93.50	31	155
Bcan5	81.23	79.78	98.92	51	345
Crscr1	89.43	88.97	99.23	319	1877
Crscr2	88.36	86.37	93.42	306	1186
Crscr3	89.28	82.24	92.04	314	1238
Crscr4	88.82	87.29	97.24	315	1250
Crscr5	89.28	86.52	95.41	316	2137

Table 6: Ruleset comprehensibility comparison (%)

	Minimum subset	Significance based
Pidia1	91.19	61.23
Pidia2	99.09	59.94
Pidia3	99.94	69.11
Pidia4	98.52	62.44
Pidia5	99.81	60.67
Bcan1	95.20	66.67
Bcan2	97.58	66.67
Bcan3	92.22	65.93
Bcan4	99.28	55.56
Bcan5	98.26	75.16
Crscr1	98.20	39.23
Crscr2	98.91	25.84
Crscr3	97.62	26.28
Crscr4	99.62	26.46
Crscr5	99.35	45.08

hence the identical ensemble accuracies in Table 4 and 5. The significance based algorithm has a threshold parameter which is user defined and in the experiment which produced the results shown in Table 4 the threshold value used was 0.1 except for Bcan2 and Bcan5 which required a 0.05 threshold to maintain a good level of fidelity.

Comparing the results in Table 4 and 5, rule accuracy and fidelity differs slightly between the two algorithms although it is important to note that the significance based rule extraction algorithm has an adjustable threshold parameter allowing fidelity to be improved if desired by the user. The main difference in the results however is in the comprehensibility measure of both algorithms which is shown in Table 6.

The results in Table 6 show that the significance based rule extraction algorithm extracts rule sets with much better comprehensibility which becomes more apparent in datasets with a large number of attributes as can be seen in the results for the credit screening dataset.

CONCLUSION

In this study we presented, demonstrated and proved a rule extraction algorithm capable of extracting rules from neural networks and neural network ensembles with respect to the users desired level of fidelity or comprehensibility. This potentially allows a broader application of neural networks and neural networks ensembles in domains where model transparency is needed. The ability to choose between fidelity and comprehensibility further allows explanations to be quickly adjusted to the needs of the user or application in which the model is being applied.

REFERENCES

Andrews, R., J. Diederich and A.B. Tichle, 1995. A survey critique of techniques for extracting rules from trained artificial neural networks. Knowledge-Based Syst., 8: 373-389.

Eriki, P.O. and R.I. Udegbunam, 2010. Application of neural network in evaluating prices of housing units in Nigeria: A preliminary investigation. J. Artif. Intell., 3: 161-167.

Khanale, P.B. and S.D. Chitnis, 2011. Handwritten devanagari character recognition using artificial neural network. J. Artif. Intell., 4: 55-62.

Khatib, T. and S. Al-Sadi, 2011. Modeling of wind speed for palestine using artificial neural network. J. Applied Sci., 11: 2634-2639.

Lotfi, A. and A. Benyettou, 2011. Using probabilistic neural networks for handwritten digit recognition. J. Artif. Intell., 4: 288-294.

Senol, D. and M. Ozturan, 2010. Stock price direction prediction using artificial neural network approach: The case of Turkey. J. Artif. Intell., 3: 261-268.

Shahrabi, J., S.S. Mousavi and M. Heydar, 2009. Supply chain demand forecasting: A comparison of machine learning techniques and traditional methods. J. Applied Sci., 9: 521-527.

Shakiba, M., M. Teshnehlab, S. Zokaie and M. Zakermoshfegh, 2008. Short-term prediction of traffic rate interval router using hybrid training of dynamic synapse neural network structure. J. Applied Sci., 8: 1534-1540.

Tahir, N.M. and H.H. Manap, 2012. Parkinson disease gait classification based on machine learning approach. J. Applied Sci., 12: 180-185.

Tanoh, A., D.K. Konan, M. Koffi, Z. Yeo, M.A. Kouacou, B.K. Koffi and K.R. N`guessan, 2008. A neural network application for diagnosis of the asynchronous machine. J. Applied Sci., 8: 3528-3531.

Tickle, A.B., M. Golea, R. Hayward and J. Diederich, 1997. The truth is in there: Current issues in extracting rules from trained feedforward artificial neural networks. Proc. Int. Conf. Neural Network, 4: 2530-2534.

Tsukimoto, H., 2000. Extracting rules from trained neural networks. IEEE Trans. Neural Networks, 11: 377-389.

Wall, R. and P. Cunningham, 2000. Exploring the potential for rule extraction from ensembles of neural networks. Proceedings of the 11th Irish Conference on Artificial Intelligence and Cognitive Science, August 23-25, 2000, National University of Ireland, Galway, Ireland, pp: 52-68.

Wall, R., P. Cunningham and W. Walsh, 2002. Explaining predictions from a neural network ensemble one at a time. Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, August 19-23, 2002, Helsinki, Finland, pp: 449-460.