Neurocomputing 72 (2009) 2845–2856

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Binary classification using ensemble neural networks and interval neutrosophic sets

Pawalai Kraipeerapun, Chun Che Fung *

School of Information Technology, Murdoch University, Perth, Australia

ARTICLE INFO

Available online 8 April 2009

Keywords:
Binary classification
Feed-forward backpropagation neural network
Interval neutrosophic sets
Uncertainty

ABSTRACT

This paper presents an ensemble neural network and interval neutrosophic sets approach to the problem of binary classification. A bagging technique is applied to an ensemble of pairs of neural networks created to predict degree of truth membership, indeterminacy membership, and false membership values in the interval neutrosophic sets. In our approach, the error and vagueness are quantified in the classification process as well. A number of aggregation techniques are proposed in this paper. We applied our techniques to the classical benchmark problems including ionosphere, pima-Indians diabetes, and liver-disorders from the UCI machine learning repository. Our approaches improve the classification performance as compared to the existing techniques which applied only to the truth membership values. Furthermore, the proposed ensemble techniques also provide better results than those obtained from only a single pair of neural networks.

1. Introduction

Neural network is one of the most popular algorithms used for binary classification. A binary neural network maps the input feature vector to the network output consisting of two classes. Hansen and Salamon [19] suggested that an ensemble of accurate and diverse neural networks gives better results and less error than a single neural network. Diversity of two classifiers is said to exist when both classifiers produce different amount of output errors based on new input data [12]. Diversity can also be described as "disagreement" of the classifiers [32]. In general, there are two stages to create an ensemble of neural networks. First, several diverse neural networks are trained. Second, the outputs obtained from the ensemble of networks are aggregated. There are various techniques to produce accurate and diverse neural networks. For example, two different initial weights used to initialize two backpropagation neural networks can produce disagreement between the networks [25]. In [24], an ensemble architecture was constructed automatically by applying different number of hidden nodes in order to find the most accurate ensemble of neural networks. On the other hand, the best diversity in the ensemble was found by using negative correlation learning and different training epochs.

Furthermore, diversity in an ensemble of neural networks can also be handled by manipulation of input data or output data. Manipulation of input data can be done in several ways such as managing the number of input features and handling training data in different ways and combinations. For instance, Zenobi and Cunningham [41] applied different feature subsets in order to create diversity in an ensemble. They also found that a diverse ensemble of less accurate classifiers outperforms an ensemble of more accurate classifiers but with less diversity. In order to handle the training data, several algorithms can be used. For example, bagging and boosting algorithms can be used to manage training data for supporting diversity in an ensemble. Bagging is based on bootstrap resampling which provides diversity by random replacement based on the original training data [9]. The objective is to derive several training sets with the replacement while maintaining the same number of training patterns as the original set. Boosting provides diversity by manipulating each training set according to the performance of the previous classifier [36]. In addition, diversity can be provided by applying artificial training samples. Melville and Mooney [32] built a different training set for each new classifier by adding artificially constructed samples to the original training data. In order to construct sample labels, they assigned the class label that disagrees with the current ensemble to the constructed sample label. An example algorithm that manipulates diversity using output data is error correcting output coding. In this algorithm, a unique codeword is created for each class label and it is used as a distributed output representation [13].

Ensemble of neural networks can improve the accuracy of classification performances. However, imperfection still exists. Imperfection always exists in real world data and also in the prediction process. In order to improve the accuracy in the binary neural network classification, assessment of imperfection in the

* Corresponding author. Tel.: +61 893607586; fax: +61 893602941.
E-mail addresses: p.kraipeerapun@murdoch.edu.au (P. Kraipeerapun), l.fung@murdoch.edu.au (C.C. Fung).

0925-2312/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.07.017

classification is an important task. The degree of imperfection may be used as an indication of the level of quality in the classification. Smets [38] categorized imperfection into three major aspects: imprecision, inconsistency, and uncertainty. He suggested that imprecision occurs if several worlds are compatible with the available information whereas inconsistency happens when there is no world agreeable to the information. Uncertainty occurs when there is a lack of information about the world for deciding if the statement is true or false. These three aspects are related to one another. For instance, imprecision in the data is a major cause of uncertainty [38]. Examples of imprecision are ambiguity, error, and vagueness. Ambiguity occurs when the decision deals with doubt. Errors can result from several sources such as measurement, data entry, or processing [17]. Vagueness deals with the concept of boundaries which cannot be defined precisely [15]. Vague objects can be categorized into three types: vague point, vague line, and vague region [14]. In this study, we consider the output of the prediction as vague point, which is a finite set of disjoint sites with known location, but the existence of the sites may be uncertain.

A variety of methods are used to deal with these causes of uncertainty. These include stochastic models, probability theory, supervaluation theory, and fuzzy logic [15,17,18,30]. These methods deal with different causes of uncertainty to different degrees of success. Although probability theory is best suited in dealing with error, stochastic models can also be used [17]. However, stochastic models are not good at handling vagueness because phenomena are assumed to be crisp [16]. Fuzzy set theory is suitable when the uncertainty arises from vagueness [17,16]. Supervaluation is also used to deal with vagueness [30]. Furthermore, neural networks can also be used to deal with errors. In [37], Generalized Regression Neural Networks (GRNN) is used to predict errors from known errors obtained from training neural networks. These predicted errors were then used as dynamic weights in the determination of results from the ensemble of neural networks.

In this paper, we start our experiment by applying two feed-forward backpropagation neural networks trained with the same input data and same parameters, but they disagree in the output target. The number of hidden neurons used in the hidden layer is one of the major issues to establish a feed-forward neural network. There are many algorithms used to determine the number of hidden neurons. For example, a neuron can be automatically created or removed from the hidden layer according to some conditions. For example, the threshold value can be compared to the training error rate [3,20], or a set of rules can be created based on the error values [6]. The result of the comparison or the satisfaction of the rules can be used to adjust the number of neurons. In [31,34], the number of hidden neurons was determined based on the evolutionary programming. In [23], Igelnik and Pao estimated the size of hidden layer on basis of bound of generalization error. In [8], the number of hidden neurons can be determined based on the analysis of variance of the input data set. In [7], the sufficient number of hidden neurons was calculated as $\lceil M/D \rceil$ where M denotes the number of training patterns and D denotes the dimension of the input vectors. On the contrary, in [5], the number of hidden neurons was computed as $2\lceil M/D \rceil$. However, $M-1$ hidden neurons were found to be sufficient in [21,35]. On the other hand, in [22], at least two dimensional hidden neurons were found to be sufficient for approximating the posteriori probability in binary classification problem with arbitrary accuracy. In our experiment, we want to concentrate on our proposed technique without varying the parameters that are not involved in our technique. Hence, the number of hidden neurons are freezed by applying two dimensional hidden neurons to all our experiments. After the neural network environment is controlled, a pair of disagreement classifiers are created. The results obtained from our technique are compared to the results obtained from a single neural network which deals only with the truth membership values. After that, we apply this technique to ensemble neural networks.

An ensemble of pairs of neural networks is created in order to improve classification performance of a single pair of networks. A bagging technique is also applied to the ensemble in order to manage diversity using input data. The number of component networks is one of the major issues when using a bagging technique. In [33], Opitz and Maclin argued that errors are much reduced if the number of component networks in an ensemble is greater than ten to fifteen. They also found that error reduction plateaus at twenty-five networks. Chawla et al. [10] suggested that improvement plateaus in the range of 30–50 classifiers. From these reports, in order to minimize error, 30 component networks are used in this paper.

Furthermore, errors and vagueness occurred in the prediction are also considered in this paper. These two causes of uncertainty are quantified in order to enhance the classification results. In our study, only errors occurred in the prediction process are considered. For vagueness, we consider the output obtained from the prediction as vague point since the input features are known but the degree of the existence of the output is uncertain. In order to represent imperfection information in the binary neural network classification, interval neutrosophic sets are used in this paper.

The rest of this paper is organized as follows. Section 2 presents the basic theory of interval neutrosophic sets. Section 3 explains the proposed techniques for the binary neural network classification with the assessment of error and vagueness. Section 4 describes the data set and the results of our experiments. Conclusions and future work are presented in Section 5.

2. Interval neutrosophic sets

In our previous papers [26–29], we combined neural networks with interval neutrosophic sets in order to classify prediction of mineral prospectivity from a set of data into deposit or barren cells. We found that an interval neutrosophic set can represent uncertainty information and supports the classification quite well. In this paper, we aim to integrate interval neutrosophic sets with neural networks in order to express uncertainty in the binary classification to the classical benchmark data and problems.

An interval neutrosophic set is an instance of a neutrosophic set, which is generalized from the concept of classical set, fuzzy set, interval valued fuzzy set, intuitionistic fuzzy set, interval valued intuitionistic fuzzy set, paraconsistent set, dialetheist set, paradoxist set, and tautological set [40]. The membership of an element in the interval neutrosophic set is expressed by three values: truth membership, indeterminacy membership, and false membership. The three memberships are independent. In some special cases, they can be dependent. In this study, the indeterminacy membership depends on both truth and false memberships. The three memberships can be any real sub-unitary subsets and can represent imprecise, incomplete, inconsistent, and uncertain information [40]. In this paper, the memberships are used to represent only uncertain information. For example, let S be an interval neutrosophic set, then $y(75, \{25, 35, 40\}, 45)$ belongs to S means that y is in S to a degree of 75%, y is uncertain to a degree of 25% or 35% or 40%, and y is not in S to a degree of 45%. This research follows the definition of interval neutrosophic sets that is defined in [39]. This definition is described below.

Let Y be a space of points (objects). An interval neutrosophic set S in Y is defined as

$$S = \{y(T_S(y), I_S(y), F_S(y))|, \quad y \in S \wedge T_S : Y \longrightarrow [0,1] \wedge I_S : Y \longrightarrow [0,1] \wedge F_S : Y \longrightarrow [0,1]\} \tag{1}$$

where T_S is the truth membership function, I_S is the indeterminacy membership function and F_S is the false membership function.

3. Proposed methodologies for binary classification

3.1. Binary classification using interval neutrosophic sets and a pair of neural networks

In this experiment, a pair of neural networks is trained to predict degree of truth membership and false membership values. The indeterminacy membership represents two causes of uncertainty, which are error and vagueness. The three memberships form an interval neutrosophic set, which is used for binary classification. Fig. 1 shows our proposed model that consists of a set of input feature vectors, two neural networks, and a process of indeterminacy estimation. The output of this model is represented in the form of an interval neutrosophic set in which each cell in the output consists of three values: truth membership, indeterminacy membership, and false membership values.

Let X be an output of the proposed model. $X = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$ where x_i is a cell at location i, and n is the total number of cells. An interval neutrosophic set A in X can be written as

$$A = \{x(T_A(x), I_A(x), F_A(x))|, \quad x \in X \wedge T_A : X \longrightarrow [0,1] \wedge I_A : X \longrightarrow [0,1] \wedge F_A : X \longrightarrow [0,1]\} \tag{2}$$

where T_A is the truth membership function, I_A is the indeterminacy membership function, and F_A is the false membership function.

The truth neural network (Truth NN) is a neural network that is trained to predict degree of the truth memberships. The falsity neural network (Falsity NN) is also a neural network with the same inputs and architecture as the truth neural network but this network is trained to predict degree of false memberships using the complement of target outputs used in the training data of the truth network. For example, if the target output used to train the truth neural network is 1, the complement of this target output is 0. Fig. 2 shows the proposed training model used for binary classification. Training errors obtained from both networks will be used to estimate uncertainty of type error in the prediction of new input data.

In the test phase, the test or unknown input data are applied to the two networks in order to predict degree of truth and false membership values. For each input pattern, the false membership

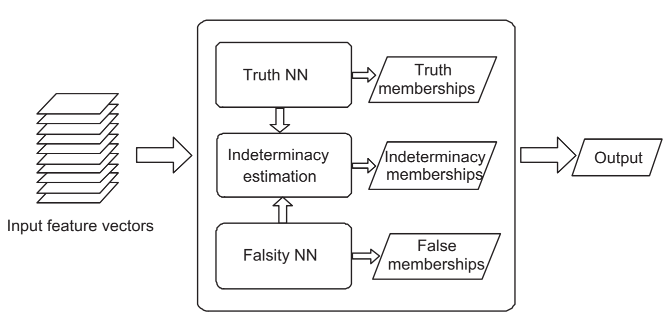

Fig. 1. The proposed binary classification based on neural networks and interval neutrosophic sets.

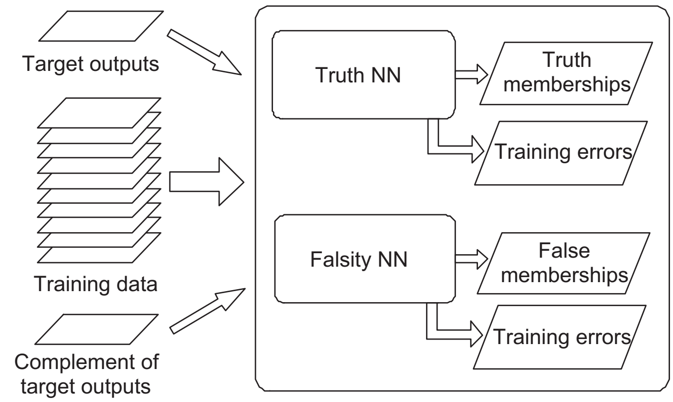

Fig. 2. The proposed training neural networks based on interval neutrosophic sets.

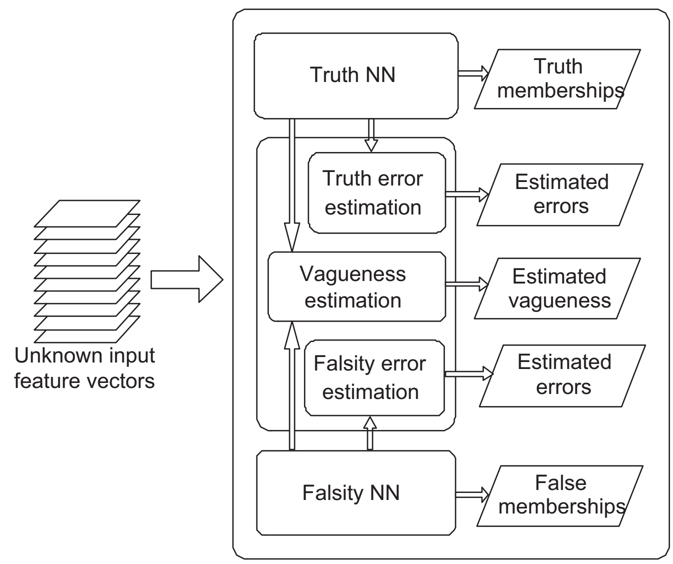

Fig. 3. The proposed model of error and vagueness estimation.

value is supposed to be the complement of the truth membership value. However, both predicted membership values may not be 100% complement to each other. Vagueness may occur in the boundary between these two memberships. Furthermore, errors may occur in the prediction of both truth and false membership values. This research deals with these two causes of uncertainty, which are vagueness and error. Fig. 3 shows our proposed model of error and vagueness estimation. The techniques used to estimate these uncertainty values are described below.

- Vagueness estimation
  In this paper, we consider the output as vague point. The input features are known but the degree of the existence of the output is uncertain. In each output pattern, the truth and false membership values are supposed to be complement to each other. If the truth membership value is 1 and the false membership value is 0 (or vice-versa) then the vagueness value is 0. However, both predicted membership values may not completely complement to each other. Vagueness may exist in the output. Fig. 4 shows the relationship among the truth membership, false membership, and vagueness values. The highest vagueness value occurs when the truth and false membership values are equal. Consequently, we compute the vagueness value as the difference between the truth membership and false membership values. If the difference

between these two values is high then the vagueness is low. On the other hand, if the difference is low then the vagueness value is high.

Let $T(x_i)$ be a truth membership at cell x_i. Let $F(x_i)$ be a false membership at cell x_i. Let $V(x_i)$ be a vagueness value at cell x_i. For each cell x_i, the vagueness value ($V(x_i)$) can be defined as follows:

$$V(x_i) = 1 - |T(x_i) - F(x_i)|. \tag{3}$$

- Error estimation

Errors can occur during training process. In this experiment, errors obtained from training process are used to estimate errors in the testing process. Fig. 5 shows our proposed error estimation technique. In order to estimate errors in the prediction of truth memberships, the known errors obtained from the truth neural network are plotted in the feature space of the input data layers. Hence, we have to deal with multidimensional space. Two methods are proposed to quantify the estimated errors. First, multidimensional interpolation [1] is used to estimate the errors. Second, scaling technique [11] is used to reduce high dimensional space into low dimensional space, and then a low dimensional interpolation method [2] is used to calculate the interpolated errors. If the multidimensional space is not too high, the first technique is suitable to apply for the interpolation. In contrast, if the multidimensional space is very high and the computer used in the experiment has a limited memory, the second technique is more suitable to apply for the interpolation. Estimated errors in the prediction of false memberships are also calculated in the same way as the estimated errors obtained from the truth neural network. The known errors obtained from training of the falsity neural network are plotted in the multidimensional feature space of the training input patterns. After that, an interpolation technique is used to estimate errors in the prediction of false memberships for the unknown input patterns.

After vagueness and error are estimated, the indeterminacy membership for each cell x_i is formed from these two types of uncertainty. Let $I(x_i)$ be an indeterminacy membership at cell x_i. For each cell x_i, the indeterminacy membership ($I(x_i)$) can be defined as follows:

$$I(x_i) = \{V(x_i), E_t(x_i), E_f(x_i)\} \tag{4}$$

where $V(x_i)$ is the vagueness value obtained from Eq. (3), $E_t(x_i)$ is the estimated error in the prediction of truth membership, and $E_f(x_i)$ is the estimated error in the prediction of false membership.

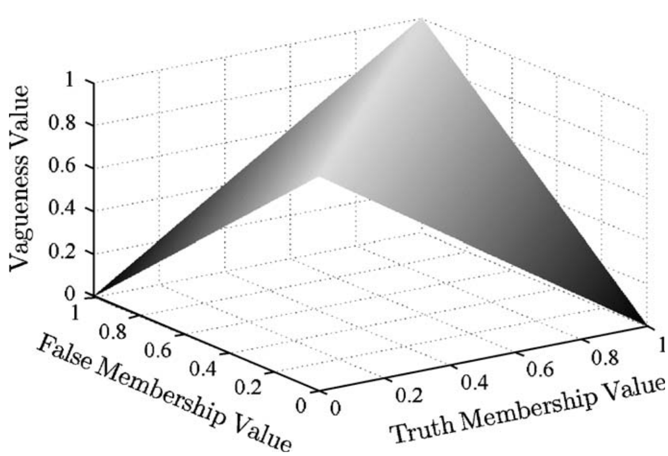

Fig. 4. The relationship among the truth membership, false membership, and vagueness values.

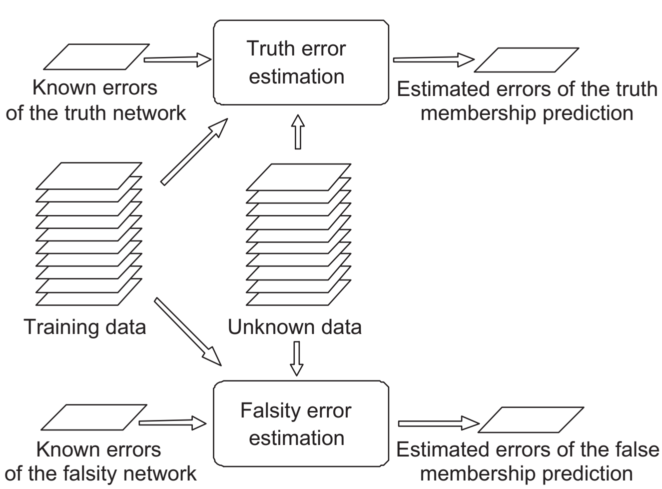

Fig. 5. The proposed error estimation technique.

After the three memberships are created in the test phase, the next step is to classify the predicted outputs into a binary class. Instead of using only the truth membership for the binary classification, the following are our proposed binary classification techniques using the truth membership, false membership, and indeterminacy membership values.

(1) Binary classification using $T > F$: For each cell x_i, if the truth membership value is greater than the false membership value ($T(x_i) > F(x_i)$) then the cell is classified as a value 1. Otherwise, the cell is classified as a value 0.

(2) Binary classification using equal weighted combination: In this method, the truth membership and the complement of the false membership for each cell are combined using a simple averaging method. The combined output $O(x_i)$ can be computed as the following:

$$O(x_i) = \frac{T(x_i) + (1 - F(x_i))}{2} \tag{5}$$

In order to classify the cell into a binary class, we apply the threshold value to classify the cell type. A range of threshold values ranging from 0.1 to 0.9 in steps of 0.1 are created and compared to the output $O(x_i)$. If the output is greater than the threshold value then the cell is classified as a value 1. Otherwise, the cell is classified as a value 0. The threshold value that can produce the best accuracy in the classification will be used in the prediction. In general, the most widely used threshold value is 0.5.

(3) Binary classification using dynamic weighted combination: In this method, uncertainty of type error is also considered in the classification. Estimated errors are used for weighting the combination between the truth and false membership values. The weight is dynamically determined based on both estimated errors: $E_t(x_i)$ and $E_f(x_i)$. The weight for the truth membership is computed as the complement of the error estimated for the truth membership. The weight for the false membership is calculated as the complement of the error estimated for the false membership. These two types of weight are considered as the certainty in the prediction of the truth and false membership values, respectively. In this study, the certainty for predicting the false membership is considered to be equal to the certainty for predicting the non-false membership value, which is the complement of the false membership value. Let $W_t(x_i)$ be the weight for the truth membership value, $W_f(x_i)$ be the weight for the false

membership value, and $W_{non-f}(x_i)$ be the weight for the non-false membership value. In this study, we consider the weight for the false membership value is equal to the weight for the non-false membership value ($W_f(x_i) = W_{non-f}(x_i)$). The dynamic combination output $O(x_i)$ can be calculated as follows:

$$O(x_i) = (W_t(x_i) \times T(x_i)) + (W_f(x_i) \times (1 - F(x_i))) \quad (6)$$

$$W_t(x_i) = \frac{1 - E_t(x_i)}{(1 - E_t(x_i)) + (1 - E_f(x_i))} \quad (7)$$

$$W_f(x_i) = \frac{1 - E_f(x_i)}{(1 - E_t(x_i)) + (1 - E_f(x_i))} \quad (8)$$

Instead of using only errors, this technique can be improved by applying both error and vagueness calculated for weighting the combination between the truth and false membership values. Hence, the average between both error and vagueness is computed and used as uncertainty in the prediction. Let $U_t(x_i)$ and $U_f(x_i)$ be the average uncertainty in the prediction of the truth and false membership values, respectively. Let $W_{tt}(x_i)$ and $W_{ff}(x_i)$ be the weight for the truth and false membership values, respectively. Therefore, the dynamic combination output $O(x_i)$ can be calculated as follows:

$$O(x_i) = (W_{tt}(x_i) \times T(x_i)) + (W_{ff}(x_i) \times (1 - F(x_i))) \quad (9)$$

$$W_{tt}(x_i) = \frac{1 - U_t(x_i)}{(1 - U_t(x_i)) + (1 - U_f(x_i))} \quad (10)$$

$$W_{ff}(x_i) = \frac{1 - U_f(x_i)}{(1 - U_t(x_i)) + (1 - U_f(x_i))} \quad (11)$$

$$U_t(x_i) = \frac{E_t(x_i) + V(x_i)}{2} \quad (12)$$

$$U_f(x_i) = \frac{E_f(x_i) + V(x_i)}{2} \quad (13)$$

Similar to the previous method, a range of threshold values are applied to the output $O(x_i)$. If the output is greater than the threshold value then the cell is classified as a value 1. Otherwise, it is classified as a value 0.

3.2. *Binary classification using interval neutrosophic sets and bagging neural networks*

In this approach, we apply interval neutrosophic sets, ensemble neural networks, and a bagging technique to the binary classification. A bagging technique is used to train neural networks in the ensemble. The bagging algorithm uses bootstrap resampling to generate multiple training sets. In this study, each bootstrap sample is created by random selection of input patterns from the original training data set with replacement. Hence, each generated training set may contain some repeated samples. Also, some original input patterns may not be included in the generated training set at all. However, each generated training set contains the same number of training patterns as the original data set. Fig. 6 shows our proposed training model based on interval neutrosophic sets, ensemble neural networks, and a bagging technique. Each component in the ensemble consists of a pair of neural networks which are the truth neural network (Truth NN) and the falsity neural network (Falsity NN). Both networks apply the same architecture and use the same generated training set for training. The truth network is trained to predict degrees of truth membership. The falsity network is trained to predict degrees of false membership. This network is trained with the complement of the target output values presented to the truth neural network. Similar to our method presented in the previous section, errors obtained from both networks can be used to estimate errors in the prediction of unknown data. Therefore, m components in the ensemble produce m pairs of truth and falsity neural networks.

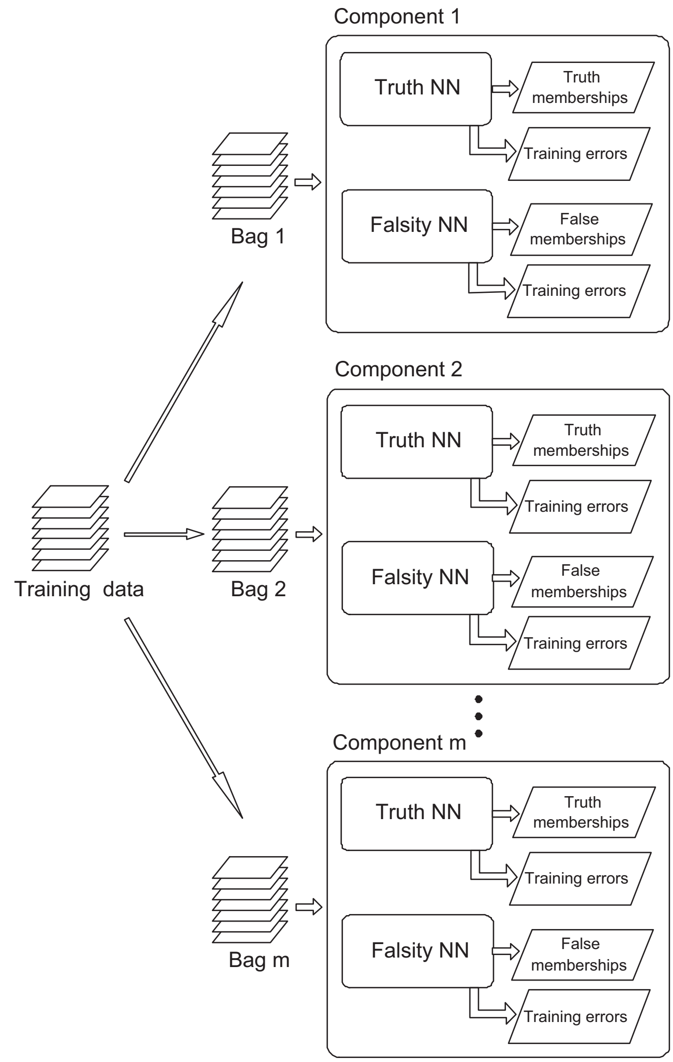

Fig. 6. The proposed training model based on interval neutrosophic sets, ensemble neural networks and a bagging technique.

In the test phase, the test data is applied to m pairs of truth and falsity networks. Each pair of the truth and falsity networks predict n pairs of the truth and false membership values, where n is the total number of test data. Fig. 7 shows our proposed binary classification model based on the integration of interval neutrosophic sets with bagging neural networks and uncertainty estimation. Similar to the previous section, vagueness may occur in the boundary between the truth and false memberships obtained from each pair of the networks in the ensemble. Errors also exist in the predictions. In order to estimate vagueness and error in the prediction, we apply the same technique as the technique applied in the previous section. Therefore, vagueness is computed from the difference between the truth and false membership values. Errors are estimated using the interpolation techniques. We consider the output from each component as an interval neutrosophic set.

Let X_j be an output of the j-th component, where $j = 1, 2, 3, \ldots, m$. Let A_j be an interval neutrosophic set in X_j. A_j

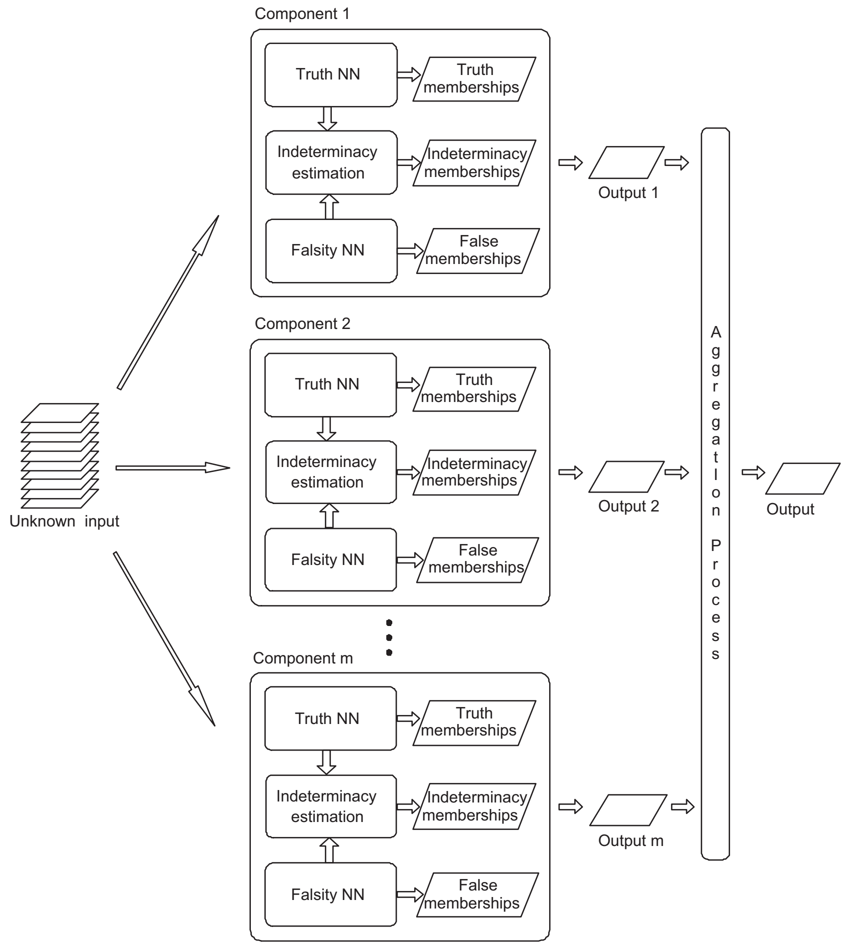

Fig. 7. The proposed binary classification model based on the integration of interval neutrosophic sets with bagging neural networks.

can be defined as

$$A_j = \{x(T_{A_j}(x), I_{A_j}(x), F_{A_j}(x))|, \quad x \in X_j \wedge T_{A_j} : X_j \longrightarrow [0,1] \wedge I_{A_j} : X_j \longrightarrow [0,1] \wedge F_{A_j} : X_j \longrightarrow [0,1]\} \tag{14}$$

$$I_j(x_i) = \{V_j(x_i), Et_j(x_i), Ef_j(x_i)\} \tag{15}$$

$$V_j(x_i) = 1 - |T_{A_j}(x_i) - F_{A_j}(x_i)| \tag{16}$$

where T_{A_j} is the truth membership function, I_{A_j} is the indeterminacy membership function, F_{A_j} is the false membership function, V_j is the vagueness value, E_{t_j} is the estimated error in the prediction of truth membership, and E_{f_j} is the estimated error in the prediction of false membership.

The next step is to combine the outputs from all components in the ensemble and then classify the cell into a binary class. In this study, we propose two methods for combining the outputs.

(1) Averaging based on truth and false memberships: Four techniques using averaging are proposed and described below.

(a) Average based on $T > F$: In this technique, the truth membership values obtained from all components are averaged. The false membership values obtained from all components are also averaged. Let $T_{avg}(x_i)$ be an average truth membership value for the cell at location i. Let $F_{avg}(x_i)$ be an average false membership value for the cell at location i. $T_{avg}(x_i)$ and $F_{avg}(x_i)$ can be defined as following:

$$T_{avg}(x_i) = \frac{\sum_{j=1}^{m} T_{A_j}(x_i)}{m} \tag{17}$$

$$F_{avg}(x_i) = \frac{\sum_{j=1}^{m} F_{A_j}(x_i)}{m} \tag{18}$$

After the average truth membership and the average false membership are computed, these two values are compared in order to classify the cell into a binary class. If the average truth membership value is greater than the average false membership value ($T_{avg}(x_i) > F_{avg}(x_i)$) then the cell is classified as a value 1. Otherwise, the cell is classified as a value 0.

(b) Average based on equal weighted combination: In this technique, the average truth membership value and the complement of the average false membership value are combined using a simple averaging technique. The combined output $O(x_i)$ can be computed as following:

$$O(x_i) = \frac{T_{avg}(x_i) + (1 - F_{avg}(x_i))}{2} \tag{19}$$

In order to classify the cell into a binary class, the threshold value is applied. A range of threshold values ranging from 0.1 to 0.9 in steps of 0.1 are created and compared to the output $O(x_i)$. If the output is greater than the threshold value then the cell is classified as a value 1. Otherwise, the cell is classified as a value 0. The threshold value that can produce the best accuracy in the classification can be used in the prediction.

(c) Dynamic weighted average based on $T>F$: In this technique, the truth membership and false membership values are weighted before averaging. The weight can be created based on members of the indeterminacy membership. In this study, we found that the weight created based on vagueness provide better results than the weight created based on only errors or both error and vagueness values. In our technique, the estimated errors are created based on the input patterns. In a bagging technique, only some input patterns are selected for training. Hence, only some known errors are used in the interpolation in order to find the interpolated errors. Therefore, the estimated errors may not give us the best results for weighting. Consequently, we create the weight based on the vagueness value ($V_j(x_i)$). Let $P(x_i)$ be an average truth membership value based on weights. Let $Q(x_i)$ be an average false membership value based on weights. $W_j(x_i)$ be the weight based on vagueness value at the j-th component. P, Q, and W can be defined as following:

$$P(x_i) = \sum_{j=1}^{m}(W_j(x_i) \times T_{A_j}(x_i)) \tag{20}$$

$$Q(x_i) = \sum_{j=1}^{m}(W_j(x_i) \times F_{A_j}(x_i)) \tag{21}$$

$$W_j(x_i) = \frac{1 - V_{A_j}(x_i)}{\sum_{j=1}^{m}(1 - V_{A_j}(x_i))} \tag{22}$$

After the dynamic weighted average truth membership value and the dynamic weighted average false membership value are computed for each input pattern, these two values are compared. If the average truth membership value is greater than the average false membership value ($P(x_i)>Q(x_i)$) then the input pattern is classified as a value 1. Otherwise, it is classified as a value 0.

(d) Dynamic weighted average based on equal weighted combination: In this technique, the dynamic weighted average truth membership value $P(x_i)$ and the complement of the dynamic weighted average false membership value $Q(x_i)$ are combined using a simple averaging technique. The combined output $O(x_i)$ can be computed as following:

$$O(x_i) = \frac{P(x_i) + (1 - Q(x_i))}{2} \tag{23}$$

After that, the combined output is compared to a range of threshold values. If the output is greater than the threshold value then the cell is classified as a value 1. Otherwise, the cell is classified as a value 0. The threshold value that can produce the best accuracy in the classification can be used in the prediction.

(2) Majority vote based on truth and false memberships: Three techniques using majority vote are proposed and described below.

(a) Majority vote based on $T>F$: In this technique, each pair of the truth and falsity neural networks produces a separate classification. For each cell x_i in the output of the j-th component, if the truth membership value is greater than the false membership value ($T_{A_j}(x_i)>F_{A_j}(x_i)$) then the cell is classified as a value 1. Otherwise, the cell is classified as a value 0. Once each cell in each output is classified, the majority vote is then applied to the ensemble outputs for each input pattern. If at least half of the outputs yield a value 1 then the cell corresponding to the input pattern is classified as a value 1. Otherwise, the cell is classified as a value 0.

(b) Majority vote based on equal weighted combination: In this technique, the truth membership and the complement of the false membership values for each cell in each output are combined using a simple averaging method. Let $O_j(x_i)$ be the combined output for the cell x_i at the j-th component. $O_j(x_i)$ can be computed as following:

$$O_j(x_i) = \frac{T_{A_j}(x_i) + (1 - F_{A_j}(x_i))}{2} \tag{24}$$

A range of threshold values are then compared to the result of the combination, $O_j(x_i)$. In this study, we found that the threshold value that frequently produces the best accuracy in the classifications is 0.5. Hence, we decided to apply the threshold value of 0.5 for the classification for all components in the ensemble. If $O_j(x_i)$ is greater than 0.5 then the cell x_i is classified as a value 1. Otherwise, the cell is classified as a value 0. After that, the majority vote is used to make a final classification. If at least half of the outputs yield a value 1 then the cell x_i is classified as a value 1. Otherwise, the cell is classified as a value 0.

(c) Majority vote based on dynamic weighted combination: In this technique, each cell x_i in the output of the j-th component is considered. Hence, we cannot use a vagueness value for weighting each pair of the truth and false membership values. Therefore, estimated errors: $Et_j(x_i)$ and $Ef_j(x_i)$ are applied in the classification. These two estimated errors are used for weighting the combination between the truth and false membership values for each cell x_i at the j-th component. The weights created for the truth and false memberships are computed as the complement of the estimated errors in the prediction of the truth and false memberships, respectively. These two types of weight are considered as the certainty in the prediction. In this study, the certainty for predicting the false membership is equal to the certainty for predicting the non-false membership value.

Let $Wt_j(x_i)$ be the weight for the truth membership value at cell x_i in the j-th component, $Wf_j(x_i)$ be the weight for the false membership value at cell x_i in the j-th component. In this technique, both weights for the false and non-false membership values are equal. The dynamic combination output $O_j(x_i)$ at cell x_i in the j-th component can be calculated as follows:

$$O_j(x_i) = (Wt_j(x_i) \times T_{A_j}(x_i)) + (Wf_j(x_i) \times (1 - F_{A_j}(x_i))) \tag{25}$$

$$Wt_j(x_i) = \frac{1 - Et_j(x_i)}{(1 - Et_j(x_i)) + (1 - Ef_j(x_i))} \tag{26}$$

$$Wf_j(x_i) = \frac{1 - Ef_j(x_i)}{(1 - Et_j(x_i)) + (1 - Ef_j(x_i))} \tag{27}$$

After that, a range of threshold values are compared to the output and then the majority vote is used to classify the cell.

4. Experiments

4.1. Data set

Three data sets from UCI Repository of machine learning data sets [4] are employed in this paper. Table 1 shows the

Table 1
Data sets used in this study.

Name	Feature type	No. of classes	No. of features	Size of samples	Size of training set	Size of testing set
Ionosphere	Numeric	2	34	351	200	151
Pima	Numeric	2	8	768	576	192
Liver	Numeric	2	6	345	276	69

characteristics of these three data sets including the size of training and testing data used in our experiments.

4.2. Experimental methodology and results

In this paper, three data sets named ionosphere, pima, and liver from UCI Repository are used to test our proposed models. Each data set is split into a training set and a testing set as shown in Table 1. In the binary classification using a single pair of neural networks, twenty pairs of feed-forward backpropagation neural networks are trained with 20 different randomized training sets in order to provide an average of 20 classification results.

In the binary classification using an ensemble of pairs of neural networks, 20 ensembles are also created in order to provide an average of 20 classification results. For each ensemble, 30 generated training sets are created using bootstrap resampling with replacement and applied to 30 components in the ensemble. For each component, a pair of feed-forward backpropagation neural networks is trained in order to predict degree of truth membership and degree of false membership values.

For each pair of neural networks in both single and ensemble techniques, the first network is used as the Truth NN whereas the second network is used as the Falsity NN. The truth network predicts degrees of truth membership. The falsity network predicts degrees of false membership. In this paper, we want to focus on our technique that aims to increase diversity by creating a pair of opposite networks. Therefore, both networks apply the same parameter values and are initialized with the same random weights. The number of input-nodes for each network is equal to the number of input features for each training set. Both networks include one hidden layer constituting of $2n$ neurons where n is the number of input features. The only difference for each pair of networks is that the target outputs of the falsity network are equal to the complement of the target outputs used to train the truth network. In order to compare all techniques created in this paper, we apply the same architecture and parameters for all pairs of neural networks.

In the test phase, after each pair of truth and false memberships is predicted, the indeterminacy memberships are then estimated. For the results obtained from a single pair of networks, Eq. (3) is used to compute the vagueness value whereas Eq. (16) is used for the results obtained from the proposed ensemble technique.

Errors in the prediction of truth and false memberships are estimated using interpolation techniques. In this paper, the multidimensional interpolation technique is applied to the liver data set. For the ionosphere and pima data sets, the scaling technique is applied in order to reduce high dimensional space into two dimensional space and then the two dimensional interpolation technique is used to estimate the interpolated errors.

After the three memberships are determined, these membership values are used for binary classification. All our proposed classification techniques explained in the previous section are then applied to the three membership values. Table 2 shows the comparison between results obtained from our proposed techniques and results obtained from the existing techniques. This table is separated into three parts. The first part shows the comparison between the average classification accuracy obtained by applying 20 pairs of truth and falsity neural networks, and the average classification accuracy obtained by applying 20 single truth neural networks. For each test set, we found that the results obtained from our proposed techniques outperform the results obtained from the existing technique, $T>0.5$. Our proposed techniques using $T>F$ and using equal weighted combination are found to provide the same results for each test set used in this study. Both techniques provide better results when compared to the existing technique. However, they give us different advantages. The technique based on $T>F$ gives us discrete results (1 or 0) whereas our technique using equal weighted combination gives us continuous results. Hence, both techniques are suitable for different applications. Our technique using equal weighted combination can be enhanced by using the dynamic weighted combination technique. In the dynamic weight combination technique, the weights can be created based on errors or based on both error and vagueness values. We found that the dynamic weight combination technique that applies both error and vagueness values provide better results than the other techniques used in the experiment of single pair of neural networks. These results show that if we know the cause of uncertainty, we can use it to improve the classification results. However, all techniques applied to a single pair of networks can be enhanced by using an ensemble of pairs of networks.

Table 2
The percentage of average classification accuracy for the test set obtained by applying our techniques and the existing techniques.

Technique	Ionosphere % correct	Pima % correct	Liver % correct
Single pair of NNs			
$T>0.5$	93.54	70.49	62.68
$T>F$	96.42	74.74	66.52
Equal weight	96.42	74.74	66.52
Dynamic weight			
Error	96.32	74.92	66.59
Error and vagueness	96.42	74.95	67.03
Ensemble of pairs of NNs and averaging			
$T>0.5$	96.56	77.16	69.93
$T>F$	97.55	77.81	74.64
Equal weight	97.55	77.81	74.64
Dynamic weight			
$T>F$	97.48	77.96	74.13
Equal weight	97.48	77.96	74.13
Ensemble of pairs of NNs and majority vote			
$T>0.5$	96.56	76.02	69.93
$T>F$	97.52	77.66	73.99
Equal weight	97.52	77.66	73.99
Dynamic weight	97.42	78.89	73.19

The second and third parts of Table 2 show the comparison between the average results obtained from 20 ensembles of pairs of neural networks and the average results obtained from 20 ensembles of the truth neural networks only. The second part shows the results based on averaging techniques whereas the third part shows the results based on majority vote techniques. From both parts, we found that the results obtained from our proposed averaging and majority vote techniques outperform the results obtained from both simple averaging and simple majority vote that apply only to the ensemble of the truth networks. In the second part, we also found that our proposed averaging technique based on $T>F$ and averaging technique based on equal weighted combination give us equal results for each test set. Also, the results obtained from both dynamic weighted average techniques are equal for each test set. In the third part, we also found that our proposed majority vote based on $T>F$ and majority vote based on equal weighted combination provide us the same results for each test set. Therefore, we can conclude that the technique of $T>F$ and the technique of weighting can be used interchangeably. Both techniques are suitable for different applications. Furthermore, we also found that our proposed averaging techniques give better classification performance compared to our proposed majority vote techniques.

A further advantage of our model is the ability to represent uncertainty in the prediction for each input pattern. This ability can be used to support the confidence in the classification. In our experiment, we found that uncertainty of type vagueness can be used to support the confidence in the classification quite well. For example, if the output pattern contains a high vagueness value then this pattern should be reconsidered. Tables 3–5 show three examples of the ranges of vagueness values in the classification of pima data set. These three examples are picked up from three out of 20 pairs of neural networks created in the first part of Table 2. For each table, vagueness values are categorized into three levels: high, med, and low. The total number of correct and incorrect outputs predicted from each pair of neural networks are represented. Table 3 shows the worst accuracy result whereas the best accuracy result is shown in Table 5. Table 4 shows the moderate accuracy result. These three tables show that most of the outputs that have low level of vagueness are correctly classified. In Table 3, the minimum vagueness value is very high which is the value of 0.6851. Hence, this classifier gives us the worst accuracy result, which is 63.54%. In contrast, the minimum vagueness value obtained from Table 5 is 0.0332 which is very low, and most of the results are fall into the group of the low level of vagueness. Therefore, this classifier provides us the best accuracy classification result, which is 80.21%.

The visualization of uncertainty of type vagueness in the classification can be used to enhance the decision making. The relationship among the truth membership, false membership, and vagueness values from Tables 3–5 can be represented in two and three dimensional spaces. Figs. 8–10 show the relationship between the truth membership and false membership values in two dimensional spaces. For the two dimensional graphical representation, if a cloud of points is arranged in the diagonal

Table 3
Total number of correct and incorrect outputs predicted from a pair of neural networks for the test set of pima data (classifier 1).

Vagueness		Number of pattern		% correct
Value	Level	Correct	Incorrect	
0.7909–0.8438	High	2	4	33.33
0.7380–0.7908	Med	15	12	55.56
0.6851–0.7379	Low	105	54	66.04
Total		122	70	63.54

Table 4
Total number of correct and incorrect outputs predicted from a pair of neural networks for the test set of pima data (classifier 2).

Vagueness		Number of pattern		% correct
Value	Level	Correct	Incorrect	
0.7611–0.9961	High	29	23	55.77
0.5261–0.7610	Med	44	25	63.77
0.2912–0.5260	Low	66	5	92.96
Total		139	53	72.40

Table 5
Total number of correct and incorrect outputs predicted from a pair of neural networks for the test set of pima data (classifier 3).

Vagueness		Number of pattern		% correct
Value	Level	Correct	Incorrect	
0.6732–0.9932	High	31	21	59.62
0.3532–0.6731	Med	47	11	81.03
0.0332–0.3531	Low	76	6	92.68
Total		154	38	80.21

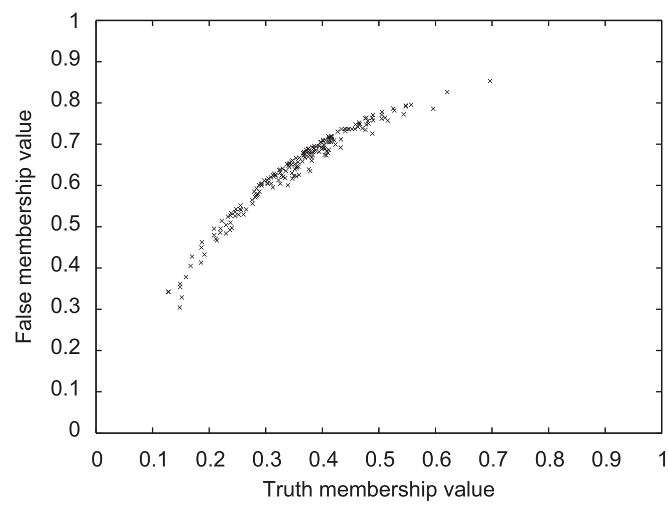

Fig. 8. Two dimensional visualization of the test set of pima data obtained from a pair of neural networks (classifier 1). The '×' represents results obtained from $T \leqslant F$.

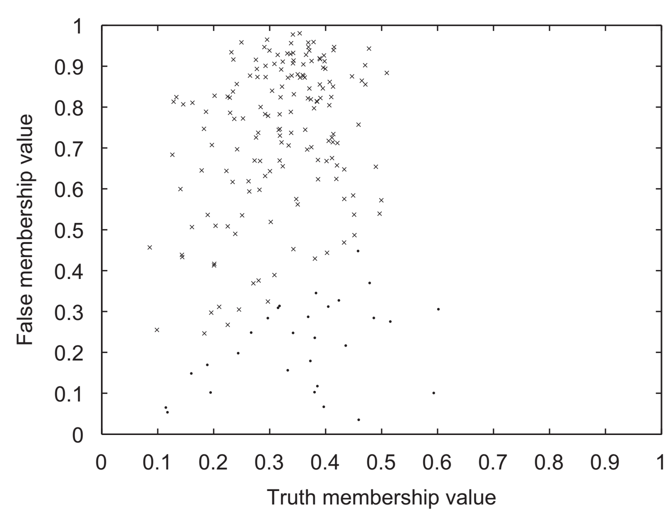

Fig. 9. Two dimensional visualization of the test set of pima data obtained from a pair of neural networks (classifier 2). The '·' represents results obtained from $T>F$ and the '×' represents results obtained from $T \leqslant F$.

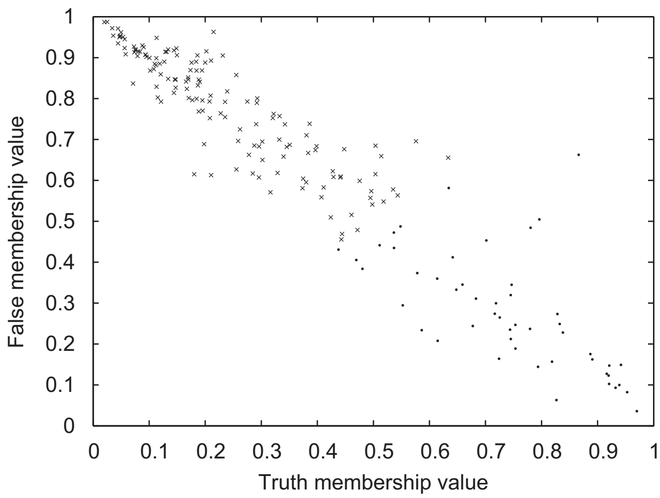

Fig. 10. Two dimensional visualization of the test set of pima data obtained from a pair of neural networks (classifier 3). The '·' represents results obtained from $T>F$ and the '×' represents results obtained from $T\leqslant F$.

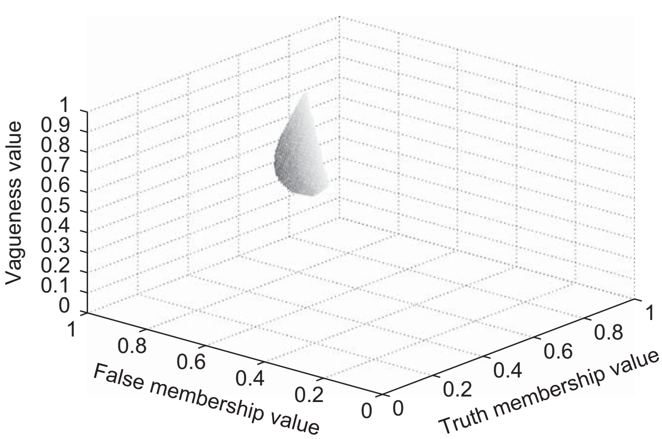

Fig. 11. Three dimensional visualization of the test set of pima data obtained from a pair of neural networks (classifier 1).

left as shown in Fig. 10 then the vagueness is low. This situation provides us a high classification accuracy result. On the other hand, the vagueness is very high if a cloud of points is arranged in the diagonal right as shown in Fig. 8. Also, this situation gives us the worse accuracy result.

Figs. 11–13 show the relationship among the truth membership, false membership, and vagueness values in three dimensional spaces. These graphical representations are displayed based on the interpolated surface of vagueness values. They are corresponding to Figs. 8–10, respectively. The vagueness level shown in these representations can be used as an indicator in order to support the decision making. The results shown in Fig. 11 represent higher uncertainty than the results shown in Fig. 12. Also, the results shown in Fig. 12 represent higher uncertainty than the results shown in Fig. 13 while the accuracy results obtained from Fig. 13 provide us the best results.

5. Conclusion and future work

In this paper, two approaches are created for binary classification. The first approach applies a single pair of neural networks whereas the second approach improved on the first one by using an ensemble of pairs of neural networks for the binary classification. Each pair of networks provides the truth and false membership values. These two values are used to compute vagueness in the prediction. Furthermore, errors occurred in the prediction are also estimated. Interpolation techniques are used to estimate those errors. Both approaches apply interval neutrosophic sets in order to represent imperfection in the prediction. In this paper, there are several techniques created based on these two approaches. The results obtained from a single pair of networks are compared to the results obtained from an ensemble of pairs of networks. The proposed ensemble technique provides better classification accuracy than the proposed single pair of networks. All our proposed techniques also improve the classification performance compared to the existing techniques using only the truth networks. Therefore, we can observe that using the complementary neural networks with the quantification of vagueness and errors can be a good starting point for further research on the binary neural network classification. In future, we will apply our techniques to cellular automata in order to represent imperfection in the classification.

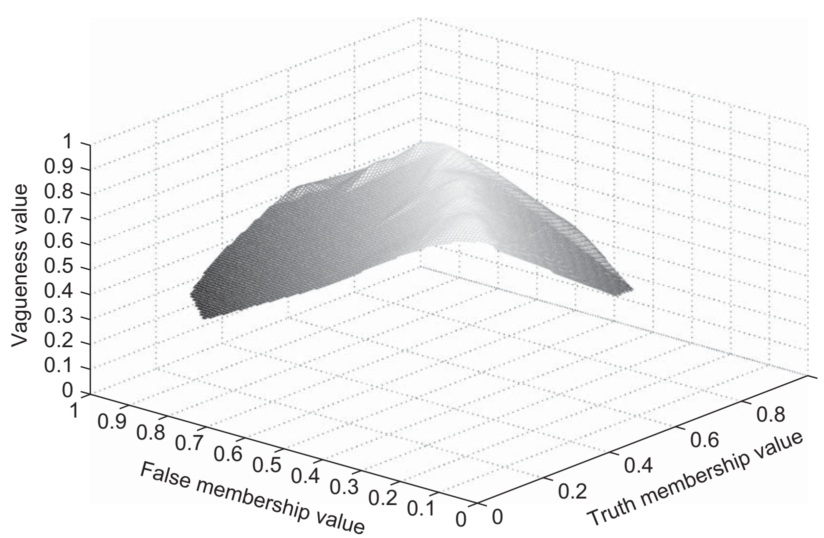

Fig. 12. Three dimensional visualization of the test set of pima data obtained from a pair of neural networks (classifier 2).

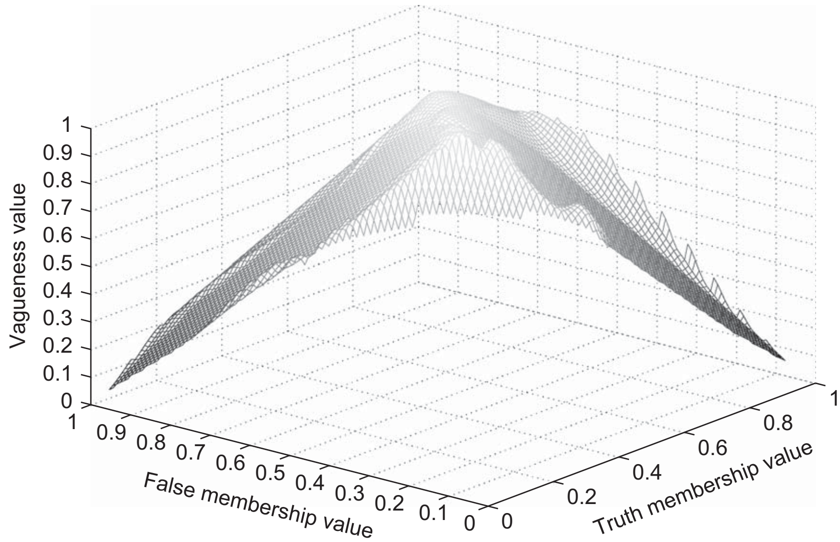

Fig. 13. Three dimensional visualization of the test set of pima data obtained from a pair of neural networks (classifier 3).

References

[1] P. Alfeld, Scattered data interpolation in three or more variables, in: Mathematical Methods in Computer Aided Geometric Design, Academic Press, New York, 1989, pp. 1–34.

[2] I. Amidror, Scattered data interpolation methods for electronic imaging systems: a survey, Journal of Electronic Imaging 11 (2002) 157–176.

[3] T. Ash, Dynamic node creation in backpropagation networks, Connection Science 1 (1989) 365–375.

[4] A. Asuncion, D. Newman, UCI machine learning repository, 2007, URL ⟨http://www.ics.uci.edu/JJsimmlearn/MLRepository.html⟩.

[5] A. Atiya, On the required size of multilayer networks for implementing real-valued functions, in: IEEE International Conference on Neural Networks, vol. 3, 1994.

[6] S. Aylward, R. Anderson, An algorithm for neural network architecture generation, in: AIAA Computing in Aerospace VIII, 1991.

[7] E. Baum, On the capabilities of multilayer perceptrons, Journal of Complexity 4 (1988) 193–215.

[8] Z. Boger, H. Guterman, Knowledge extraction from artificial neural network models, in: IEEE International Conference on Systems, Man, and Cybernetics, SMC97, vol. 4, 1997.

[9] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.

[10] N. Chawla, T. Moore, K. Bowyer, L. Hall, C. Springer, W. Kegelmeyer, Bagging-like effects for decision trees and neural nets in protein secondary structure prediction, in: ACM SIGKDD Workshop on Data Mining in Bio-Informatics, 2001.

[11] T.F. Cox, M.A.A. Cox, Multidimensional Scaling, second ed., Chapman & Hall, CRC Press, London, Boca Raton, FL, 2001.

[12] T.G. Dietterich, Ensemble methods in machine learning, in: J. Kittler, F. Roli (Eds.), Multiple Classifier Systems, Lecture Notes in Computer Science, vol. 1857, Springer, Berlin, 2000.

[13] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research 2 (1995) 263–286.

[14] A. Dilo, P. Kraipeerapun, W. Bakker, R. de By, Storing and handling vague spatial objects, in: The 15th International Workshop on Database and Expert Systems Applications, DEXA 2004, 2004.

[15] M. Duckham, Uncertainty and geographic information: computational and critical convergence, in: Representation in a Digital Geography, Wiley, New York, 2002.

[16] M. Duckham, J. Sharp, Representing GIS, chap. Uncertainty and geographic information: Computational and critical convergence, John Wiley, New York, 2005, pp. 113–124.

[17] P.F. Fisher, Models of uncertainty in spatial data, in: Geographical Information Systems: Principles, Techniques, Management and Applications, second ed., Wiley, Chichester, 2005, pp. 69–83.

[18] P. Hájek, L. Godo, F. Esteva, Fuzzy logic and probability, in: Eleventh Annual Conference on Uncertainty in Artificial Intelligence, Montreal, Quebec, Canada, 1995.

[19] L.K. Hansen, P. Salamon, Pattern analysis and machine intelligence, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, 1990.

[20] Y. Hirose, I. Yamashita, S. Hijiya, Back-propagation algorithm which varies the number of hidden units, Neural Networks 4 (1991) 61–66.

[21] S.-C. Huang, Y.-F. Huang, Bounds on the number of hidden neurons in multilayer perceptrons, IEEE Transactions on Neural Networks 2 (1) (1991) 47–55.

[22] K. ichi Funahashi, Multilayer neural networks and Bayes decision theory, Neural Networks 11 (2) (1998) 209–213.

[23] B. Igelnik, Y.-H. Pao, Estimation of size of hidden layer on basis of bound of generalization error, in: Proceedings of IEEE International Conference on Neural Networks, vol. 4, 1995.

[24] M.M. Islam, X. Yao, K. Murase, A constructive algorithm for training cooperative neural network ensembles, IEEE Transactions on Neural Networks 14 (4) (2003) 820–834.

[25] J.F. Kolen, J.B. Pollack, Back propagation is sensitive to initial conditions, in: R.P. Lippmann, J.E. Moody, D.S. Touretzky (Eds.), Advances in Neural Information Processing Systems, vol. 3, Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1991.

[26] P. Kraipeerapun, C.C. Fung, W. Brown, Assessment of uncertainty in mineral prospectivity prediction using interval neutrosophic set, in: Y. Hao, J. Liu, Y. Wang, Y. ming Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), CIS (2), Lecture Notes in Computer Science, vol. 3802, Springer, Berlin, 2005.

[27] P. Kraipeerapun, C.C. Fung, W. Brown, K.W. Wong, Mineral prospectivity prediction using interval neutrosophic sets, in: V. Devedzic (Ed.), Artificial Intelligence and Applications, IASTED/ACTA Press, 2006.

[28] P. Kraipeerapun, C.C. Fung, W. Brown, K.W. Wong, Neural network ensembles using interval neutrosophic sets and bagging for mineral prospectivity prediction and quantification of uncertainty, in: Proceedings of the 2006 IEEE International Conferences on Cybernetics and Intelligent Systems, Bangkok, Thailand, 2006.

[29] P. Kraipeerapun, K.W. Wong, C.C. Fung, W. Brown, Quantification of uncertainty in mineral prospectivity prediction using neural network ensembles and interval neutrosophic sets, in: Proceedings of the 2006 IEEE World Congress on Computational Intelligence: A Joint Conference of the International Joint Conference on Neural Networks, IJCNN 2006, Vancouver, Canada, 2006.

[30] L. Kulik, Vague spatial reasoning based on supervaluation, in: S. Winter (Ed.), Geographical Domain and Geographical Information Systems—GeoInfo Series, Institute for Geoinformation, Vienna University of Technology, Vienna, vol. 19, 2000.

[31] J.R. McDonnell, D. Waagen, Determining neural network hidden layer size using evolutionary programming, Professional Paper ADA273242, Naval Command Control and Ocean Surveillance Center RDT and E Division, San Diego, CA, 1993.

[32] P. Melville, R.J. Mooney, Constructing diverse classifier ensembles using artificial training examples, in: Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI, 2003.

[33] D. Opitz, R. Maclin, Popular ensemble methods: an empirical study, Journal of Artificial Intelligence Research 11 (1999) 169–198.

[34] K. Peng, S.S. GE, C. Wen, An algorithm to determine neural network hidden layer size and weight coefficients, in: Proceedings of the 15th IEEE international Symposium on Intelligent Control, ISIC 2000, 2000.

[35] M.A. Sartori, P.J. Antsaklis, A simple method to derive bounds on the size and to train multilayer neural networks, IEEE Transactions on Neural Networks 2 (4) (1991) 467–471.

[36] H. Schwenk, Y. Bengio, Boosting neural networks, Neural Computation 12 (8) (2000) 1869–1887.

[37] Z.-Q. Shen, F.-S. Kong, Dynamically weighted ensemble neural networks for regression problems, in: Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, vol. 6, 2004.
[38] P. Smets, Imperfect information: imprecision-uncertainty, in: Uncertainty Management in Information Systems: From Needs to Solutions, Kluwer Academic Publishers, Dordrecht, 1997, pp. 225–254.
[39] H. Wang, D. Madiraju, Y.-Q. Zhang, R. Sunderraman, Interval neutrosophic sets, International Journal of Applied Mathematics and Statistics 3 (2005) 1–18.
[40] H. Wang, F. Smarandache, Y.-Q. Zhang, R. Sunderraman, Interval Neutrosophic Sets and Logic: Theory and Applications in Computing, Neutrosophic Book Series, No.5, ⟨http://arxiv.org/abs/cs/0505014⟩, 2005.
[41] G. Zenobi, P. Cunningham, Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error, in: L.D. Raedt, P.A. Flach (Eds.), ECML, Lecture Notes in Computer Science, vol. 2167, Springer, Berlin, 2001.

Pawalai Kraipeerapun received her B.Sc. degree in Mathematics from Chulalongkorn University, Thailand in 1993. She received her M.Sc. degree in Information Technology from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand in 1997 and M.Sc. degree in Geoinformatics (with distinction) from International Institute for Geo-Information Science and Earth Observation (ITC), The Netherlands in 2004. She is currently a Ph.D. candidate in Information Technology at Murdoch University, Western Australia. Her research interests are computational intelligence techniques and intelligent systems applications.

Chun Che Fung received the B.Sc. (Hon.) and M.Eng. degrees from the University of Wales in 1981 and 1982, respectively. He was awarded his Ph.D. degree from the University of Western Australia in 1994. He has been an Associate Professor of the School of Information Technology, Murdoch University since 2003. Prior to his present appointment, he worked at the School of Electrical and Computer Engineering, Curtin University from 1988 to 2002, and the Department of Electronic and Communication Engineering, Singapore Polytechnic from 1982 to 1988. His research interests are computational intelligence techniques and intelligent systems applications. He is a Senior Member of IEEE and a Member of ACS, IEAust and ACM.