Reliable Probabilistic Prediction for Medical Decision Support

Harris Papadopoulos[1,2]

[1] Frederick Research Center,
7-9 Filokyprou St., Palouriotisa, Nicosia 1036, Cyprus
[2] Computer Science and Engineering Department, Frederick University,
7 Y. Frederickou St., Palouriotisa, Nicosia 1036, Cyprus
h.papadopoulos@frederick.ac.cy

Abstract. A major drawback of most existing medical decision support systems is that they do not provide any indication about the uncertainty of each of their predictions. This paper addresses this problem with the use of a new machine learning framework for producing valid probabilistic predictions, called Venn Prediction (VP). More specifically, VP is combined with Neural Networks (NNs), which is one of the most widely used machine learning algorithms. The obtained experimental results on two medical datasets demonstrate empirically the validity of the VP outputs and their superiority over the outputs of the original NN classifier in terms of reliability.

Keywords: Venn Prediction, Probabilistic Classification, Multiprobability Prediction, Medical Decision Support

1 Introduction

Medical decision support is an area in which the machine learning community has conducted extensive research that resulted in the development of several diagnostic and prognostic systems [8, 11]. These systems learn to predict the most likely diagnosis of a new patient based on a past history of patients with known diagnoses. The most likely diagnosis however, is the only output most such systems produce. They do not provide any further information about how much one can trust the provided diagnosis. This is a significant disadvantage in a medical setting where some indication about the likelihood of each diagnosis is of paramount importance [7].

A solution to this problem was given by a recently developed machine learning theory called *Conformal Prediction* (CP) [24]. CP can be used for extending traditional machine learning algorithms and developing methods (called Conformal Predictors) whose predictions are guaranteed to satisfy a given level of confidence without assuming anything more than that the data are independently and identically distributed (i.i.d.). More specifically, CPs produce as their predictions a set containing all the possible classifications needed to satisfy the required confidence level. To date many different CPs have been developed, see e.g. [14,

15, 17–19, 21, 22], and have been applied successfully to a variety of important medical problems such as [3, 6, 9, 10, 16].

This paper focuses on an extension of the original CP framework, called Venn Prediction (VP), which can be used for making *multiprobability predictions*. In particular multiprobability predictions are a set of probability distributions for the true classification of the new example. In effect this set defines lower and upper bounds for the conditional probability of the new example belonging to each one of the possible classes. These bounds are guaranteed (up to statistical fluctuations) to contain the corresponding true conditional probabilities. Again, like with CPs, the only assumption made for obtaining this guaranty is that the data are i.i.d.

The main aim of this paper is to propose a Venn Predictor based on Neural Networks (NNs) and evaluate its performance on medical tasks. The choice of NNs as basis for the proposed method was made due to their successful application to many medical problems, see e.g. [1, 2, 12, 20], as well as their popularity among machine learning techniques for almost any type of application. The experiments performed examine on one hand the empirical validity of the probability bounds produced by the proposed method and on the other hand compare them with the probabilistic outputs of the original NN classifier.

The rest of this paper starts with an overview of the Venn Prediction framework in the next section, while in Section 3 it details the proposed Neural Network Venn Predictor algorithm. Section 4 presents the experiments performed on two medical datasets and reports the obtained results. Finally, Section 5 gives the conclusions and future directions of this work.

2 The Venn Prediction Framework

This section gives a brief description of the Venn prediction framework; for more details the interested reader is referred to [24]. We are given a training set $\{(x_1, y_1), \ldots, (x_l, y_l)\}$ of examples, where each $x_i \in \mathbb{R}^d$ is the vector of attributes for example i and $y_i \in \{Y_1, \ldots, Y_c\}$ is the classification of that example. We are also given a new unclassified example x_{l+1} and our task is to predict the probability of this new example belonging to each class $Y_j \in \{Y_1, \ldots, Y_c\}$ based only on the assumption that all $(x_i, y_i), i = 1, 2, \ldots$ are generated independently by the same probability distribution (i.i.d.).

The main idea behind Venn prediction is to divide all examples into a number of categories and calculate the probability of x_{l+1} belonging to each class $Y_j \in \{Y_1, \ldots, Y_c\}$ as the frequency of Y_j in the category that contains it. However, as we don't know the true class of x_{l+1}, we assign each one of the possible classes to it in turn and for each assigned classification Y_k we calculate a probability distribution for the true class of x_{l+1} based on the examples

$$\{(x_1, y_1), \ldots, (x_l, y_l), (x_{l+1}, Y_k)\}. \quad (1)$$

To divide each set (1) into categories we use what we call a *Venn taxonomy*. A Venn taxonomy is a finite measurable partition of the space of examples. Typically each taxonomy is based on a traditional machine learning algorithm, called

the *underlying algorithm* of the Venn predictor. The output of this algorithm for each attribute vector $x_i, i = 1, \ldots, l+1$ after being trained either on the whole set (1), or on the set resulting after removing the pair (x_i, y_i) from (1), is used to assign (x_i, y_i) to one of a predefined set of categories. For example, a Venn taxonomy that can be used with every traditional algorithm puts in the same category all examples that are assigned the same classification by the underlying algorithm. The Venn taxonomy used in this work is defined in the next section.

After partitioning (1) into categories using a Venn taxonomy, the category T_{new} containing the new example (x_{l+1}, Y_k) will be nonempty as it will contain at least this one example. Then the empirical probability of each label Y_j in this category will be

$$p^{Y_k}(Y_j) = \frac{|\{(x^*, y^*) \in T_{new} : y^* = Y_j\}|}{|T_{new}|}. \tag{2}$$

This is a probability distribution for the label of x_{l+1}. After assigning all possible labels to x_{l+1} we get a set of probability distributions that compose the multiprobability prediction of the Venn predictor $P_{l+1} = \{p^{Y_k} : Y_k \in \{Y_1, \ldots, Y_c\}\}$. As proved in [24] the predictions produced by any Venn predictor are automatically valid multiprobability predictions. This is true regardless of the taxonomy of the Venn predictor. Of course the taxonomy used is still very important as it determines how efficient, or informative, the resulting predictions are. We want the diameter of multiprobability predictions and therefore their uncertainty to be small and we also want the predictions to be as close as possible to zero or one.

The maximum and minimum probabilities obtained for each class Y_j define the interval for the probability of the new example belonging to Y_j:

$$\left[\min_{k=1,\ldots,c} p^{Y_k}(Y_j), \max_{k=1,\ldots,c} p^{Y_k}(Y_j)\right]. \tag{3}$$

If the lower bound of this interval is denoted as $L(Y_j)$ and the upper bound is denoted as $U(Y_j)$, the Venn predictor finds

$$j_{best} = \arg \max_{j=1,\ldots,c} L(Y_j) \tag{4}$$

and outputs the class $\hat{y} = Y_{j_best}$ as its prediction together with the interval

$$[L(\hat{y}), U(\hat{y})] \tag{5}$$

as the probability interval that this prediction is correct. The complementary interval

$$[1 - U(\hat{y}), 1 - L(\hat{y})] \tag{6}$$

gives the probability that $\hat{y}$ is not the true classification of the new example and it is called the *error probability interval*.

3 Venn Prediction with Neural Networks

This section describes the proposed Neural Network (NN) based Venn Prediction algorithm. In this work we are interested in binary classification problems ($Y_j \in \{0,1\}$) and therefore this algorithm is designed for this type of problems. The NNs used were 2-layer fully connected feed-forward networks with tangent sigmoid hidden units and a single logistic sigmoid output unit. They were trained with the scaled conjugate gradient algorithm minimizing cross-entropy error (log loss). As a result their outputs can be interpreted as probabilities for class 1 and they can be compared with those produced by the Venn predictor.

The outputs produced by a binary classification NN can be used to define an appropriate Venn taxonomy. After assigning each classification $Y_k \in \{0,1\}$ to the new example x_{l+1} we train the underlying NN on the set (1) and then input the attribute vector of each example in (1) as a test pattern to the trained NN to obtain the outputs $o_1, \ldots, o_{l+1}$. These output values can now be used to divide the examples into categories. In particular, we expect that the examples for which the NN gives similar output will have a similar likelihood of belonging to class 1. We therefore split the range of the NN output $[0,1]$ to a number of equally sized regions λ and assign the examples whose output falls in the same region to the same category. In other words each one of these λ regions defines one category of the taxonomy.

Using this taxonomy we divide the examples into categories for each assumed classification $Y_k \in \{0,1\}$ of x_{l+1} and follow the process described in Section 2 to calculate the outputs of the Neural Network Venn Predictor (NN-VP). Algorithm 1 presents the complete NN-VP algorithm.

Algorithm 1: Neural Networks Venn Predictor

Input: training set $\{(x_1, y_1), \ldots, (x_l, y_l)\}$, new example x_{l+1}, number of categories λ.
for $k = 0$ **to** 1 **do**
  Train the NN on the extended set $\{(x_1, y_1), \ldots, (x_l, y_l), (x_{l+1}, k)\}$;
  Supply the input patterns $x_1, \ldots, x_{l+1}$ to the trained NN to obtain the outputs $o_1, \ldots, o_{l+1}$;
  for $i = 1$ **to** λ **do**
    Find all examples with NN output between $(i-1)/\lambda$ and i/λ and assign them to category T_i;
  end
  Find the category $T_{new} \in \{T_1, \ldots, T_\lambda\}$ that contains (x_{l+1}, k);
  $p^k(1) := \frac{|\{(x^*, y^*) \in T_{new} : y^* = 1\}|}{|T_{new}|}$;
  $p^k(0) := 1 - p^k(1)$;
end
$L(0) := \min_{k=0,1} p^k(0)$; and $L(1) := \min_{k=0,1} p^k(1)$;
Output:
  Prediction $\hat{y} = \arg\max_{j=0,1} L(j)$;
  The probability interval for $\hat{y}$: $[\min_{k=0,1} p^k(\hat{y}), \max_{k=0,1} p^k(\hat{y})]$.

4 Experiments and Results

Experiments were performed on two medical datasets from the UCI Machine Learning Repository [5]:

- **Mammographic Mass**, which is concerned with the discrimination between benign and malignant mammographic masses based on 3 BI-RADS attributes (mass shape, margin and density) and the patient's age [4]. It consists of 961 cases of which 516 are benign and 445 are malignant.
- **Pima Indians Diabetes**, which is concerned with forecasting the onset of diabetes mellitus in a high-risk population of Pima Indians [23]. It consists of 768 cases of which 500 tested positive for diabetes. Each case is described by 8 attributes.

In the case of the Mammographic Mass data the mass density attribute was not used as it did not seem to have any positive impact on the results. Furthermore all cases with missing attribute values were removed and the 2 nominal attributes (mass shape and margin) were converted to a set of binary attributes, one for each nominal value; for each case the binary attribute corresponding to the nominal value of the attribute was set to 1 while all others were set to 0. The resulting dataset consisted of 830 examples described by 10 attributes each.

The NNs used consisted of 4 hidden units for the Mammographic Mass data, as this was the number of units used in [4], and 10 for the Pima Indians Diabetes data, as this seemed to have the best performance with the original NN classifier. All NNs were trained with the scaled conjugate gradient algorithm minimizing cross-entropy error and early stopping based on a validation set consisting of 20% of the corresponding training set. In an effort to avoid local minima each NN was trained 5 times with different random initial weight values and the one that performed best on the validation set was selected for being applied to the test examples. Before each training session all attributes were normalised setting their mean value to 0 and their standard deviation to 1. The number of categories λ of NN-VP was set to 6, which seems to be the best choice for small to moderate size datasets.

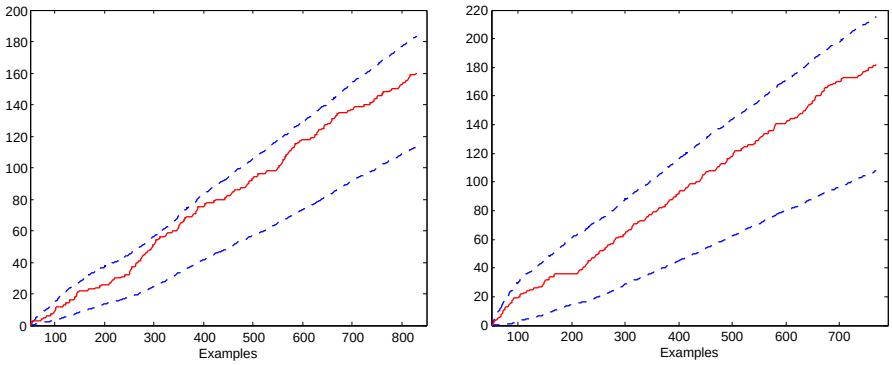
4.1 On-line Experiments

This subsection demonstrates the empirical validity of the Neural Networks Venn Predictor (NN-VP) by applying it to the two datasets in the on-line mode. More specifically, starting with an initial training set consisting of 50 examples, each subsequent example is predicted in turn and then its true classification is revealed and it is added to the training set for predicting the next example. Figure 1 shows the following three curves for each dataset:

- the cumulative error curve

$$E_n = \sum_{i=1}^{n} err_i, \tag{7}$$

  where $err_i = 1$ if the prediction $\hat{y}_i$ is wrong and $err_i = 0$ otherwise,



(a) Mammographic Mass (b) Pima Indians Diabetes

Fig. 1. On-line performance of NN-VP on the two datasets. Each plot shows the cumulative number of errors E_n with a solid line and the cumulative lower and upper error probability curves LEP_n and UEP_n with dashed lines.

– the cumulative lower error probability curve (see (6))

$$LEP_n = \sum_{i=1}^{n} 1 - U(\hat{y}_i) \tag{8}$$

– and the cumulative upper error probability curve

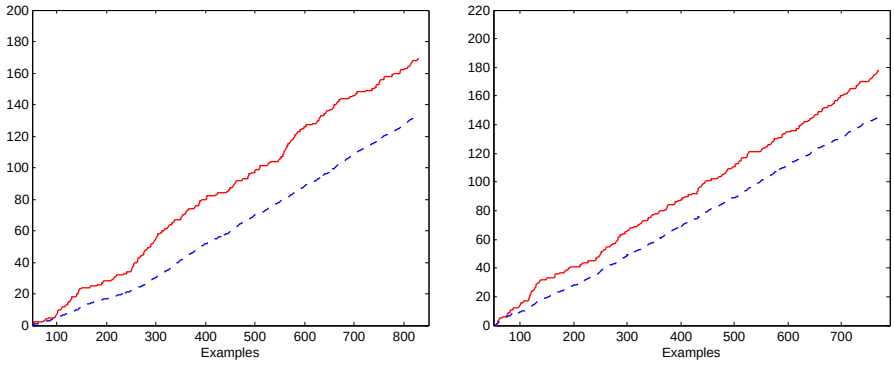
$$UEP_n = \sum_{i=1}^{n} 1 - L(\hat{y}_i). \tag{9}$$

Both plots confirm that the probability intervals produced by NN-VP are well-calibrated. The cumulative errors are always included inside the cumulative upper and lower error probability curves produced by the NN-VP.

Two analogous plots generated by applying the original NN classifier to the two datasets are shown in Figure 2. In this case the cumulative error curve (7) for each NN is plotted together with the cumulative error probability curve

$$EP_n = \sum_{i=1}^{n} |\hat{y}_i - \hat{p}_i|, \tag{10}$$

where $\hat{y}_i \in \{0, 1\}$ is the NN prediction for example i and $\hat{p}_i$ is the probability given by NN for example i belonging to class 1. In effect this curve is a sum of the probabilities of the less likely classes for each example according to the NN. One would expect that this curve would be very near the cumulative error curve if the probabilities produced by the NN were well-calibrated. The two plots of Figure 2 show that this is not the case. The NNs underestimate the true error probability in both cases since the cumulative error curve is much higher than the



(a) Mammographic Mass (b) Pima Indians Diabetes

Fig. 2. On-line performance of the original NN classifier on the two datasets. Each plot shows the cumulative number of errors E_n with a solid line and the cumulative error probability curve EP_n with a dashed line.

cumulative error probability curve. To confirm this, the p-value of obtaining the resulting total number of errors E_N by a Poisson binomial distribution with the probabilities produced by the NN was calculated for each dataset. The resulting p-values were 0.000179 and 0.000548 respectively. This demonstrates the need for probability intervals as opposed to single probability values as well as that the probabilities produced by NNs can be very misleading.

4.2 Batch Experiments

This subsection examines the performance of NN-VP in the batch setting and compares its results with those of the direct predictions made by the original NN classifier. For these experiments both datasets were divided randomly into a training set consisting of 200 examples and a test set with all the remaining examples. In order for the results not to depend on a particular division into training and test sets, 10 different random divisions were performed and all results reported here are over all 10 test sets.

Since NNs output a single probabilistic output for the true class of each example being 1, in order to compare this output with that of NN-VP the latter was converted to the mean of $L(1)$ and $U(1)$; corresponding to the estimate of NN-VP about the probability of each test example belonging to class 1. For reporting these results four quality metrics are used. The first is the accuracy of each classifier, which does not take into account the probabilistic outputs produced, but it is a typical metric for assessing the quality of classifiers. The second is cross-entropy error (log loss):

$$CE = -\sum_{i=1}^{N} y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i), \tag{11}$$

Table 1. Results of the original NN and NN-VP on the Mammographic Mass dataset

	Accuracy	CE	BS	REL
Original NN	78.83%	3298	0.1596	0.0040
NN-VP	78.92%	3054	0.1555	0.0023
Improvement (%)	0.11	7.40	2.57	42.50

Table 2. Results of the original NN and NN-VP on the Pima Indians Diabetes dataset

	Accuracy	CE	BS	REL
Original NN	74.56%	3084	0.1760	0.0074
NN-VP	74.26%	3014	0.1753	0.0035
Improvement (%)	-0.40	2.27	0.40	52.70

where N is the number of examples and $\hat{p}_i$ is the probability produced by the algorithm for class 1; this is the error minimized by the training algorithm of the NNs on the training set. The third metric is the Brier score:

$$BS = \frac{1}{N}\sum_{i=1}^{N}(\hat{p}_i - y_i)^2. \tag{12}$$

The Brier score can be decomposed into three terms interpreted as the uncertainty, reliability and resolution of the probabilities, by dividing the range of probability values into a number of intervals K and representing each interval $k = 1, \ldots, K$ by a 'typical' probability value r_k [13]. The fourth metric used here is the reliability term of this decomposition:

$$REL = \frac{1}{N}\sum_{k=1}^{K} n_k(r_k - \phi_k)^2, \tag{13}$$

where n_k is the number of examples with output probability in the interval k and ϕ_k is the percentage of these examples that belong to class 1. Here the number of categories K was set to 20.

Tables 1 and 2 present the results of the original NN and NN-VP on each dataset respectively. With the exception of the accuracy on the Pima Indians Diabetes dataset the NN-VP performs better in all other cases. Although the difference between the two methods on the first three metrics is relatively small, the improvement achieved by the VP in terms of reliability is significant. Reliability is the main concern of this work, as if the probabilities produced by some algorithm are not reliable, they are not really useful.

5 Conclusions

This paper presented a Venn Predictor based on Neural Networks. Unlike the original NN classifiers VP produces probability intervals for each of its predictions, which are valid under the general i.i.d. assumption. The experiments performed in the online setting demonstrated the validity of the probability intervals produced by the proposed method and their superiority over the single probabilities produced by NN, which can be significantly different from the observed frequencies. Moreover, the comparison performed in the batch setting showed that even when one discards the interval information produced by NN-VP by taking the mean of its multiprobability predictions these are still much more reliable than the probabilities produced by NNs.

An immediate future direction of this work is the definition of a Venn taxonomy based on NNs for multilabel classification problems and experimentation with the resulting VP. Furthermore, the application of VP to other challenging problems and evaluation of the results is also of great interest.

Acknowledgments. The author is grateful to Professors V. Vovk and A. Gammerman for useful discussions. This work was co-funded by the European Regional Development Fund and the Cyprus Government through the Cyprus Research Promotion Foundation "DESMI 2009-2010" (ORIZO/0609(BIE)/24).

References

1. Anagnostou, T., Remzi, M., Djavan, B.: Artificial neural networks for decision-making in urologic oncology. Review in Urology 5(1), 15–21 (2003)
2. Anastassopoulos, G.C., Iliadis, L.S.: Ann for prognosis of abdominal pain in childhood: Use of fuzzy modelling for convergence estimation. In: Proceedings of the 1st International Workshop on Combinations of Intelligent Methods and Applications. pp. 1–5 (2008)
3. Bellotti, T., Luo, Z., Gammerman, A., Delft, F.W.V., Saha, V.: Qualified predictions for microarray and proteomics pattern diagnostics with confidence machines. International Journal of Neural Systems 15(4), 247–258 (2005)
4. Elter, M., Schulz-Wendtland, R., Wittenberg, T.: The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process. Medical Physics 34(11), 4164–4172 (2007)
5. Frank, A., Asuncion, A.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml`
6. Gammerman, A., Vovk, V., Burford, B., Nouretdinov, I., Luo, Z., Chervonenkis, A., Waterfield, M., Cramer, R., Tempst, P., Villanueva, J., Kabir, M., Camuzeaux, S., Timms, J., Menon, U., Jacobs, I.: Serum proteomic abnormality predating screen detection of ovarian cancer. The Computer Journal 52(3), 326–333 (2009)
7. Holst, H., Ohlsson, M., Peterson, C., Edenbrandt, L.: Intelligent computer reporting 'lack of experience': a confidence measure for decision support systems. Clinical Physiology 18(2), 139–147 (1998)
8. Kononenko, I.: Machine learning for medical diagnosis: History, state of the art and perspective. Artificial Intelligence in Medicine 23(1), 89–109 (2001)

9. Lambrou, A., Papadopoulos, H., Gammerman, A.: Reliable confidence measures for medical diagnosis with evolutionary algorithms. IEEE Transactions on Information Technology in Biomedicine 15(1), 93–99 (2011)
10. Lambrou, A., Papadopoulos, H., Kyriacou, E., Pattichis, C.S., Pattichis, M.S., Gammerman, A., Nicolaides, A.: Assessment of stroke risk based on morphological ultrasound image analysis with conformal prediction. In: Proceedings of the 6th IFIP International Conference on Artificial Intelligence Appications and Innovations (AIAI 2010). IFIP AICT, vol. 339, pp. 146–153. Springer (2010)
11. Lisboa, P.J.G.: A review of evidence of health benefit from artificial neural networks in medical intervention. Neural Networks 15(1), 11–39 (2002)
12. Mantzaris, D., Anastassopoulos, G., Iliadis, L., Kazakos, K., Papadopoulos, H.: A soft computing approach for osteoporosis risk factor estimation. In: Proceedings of the 6th IFIP International Conference on Artificial Intelligence Appications and Innovations (AIAI 2010). IFIP AICT, vol. 339, pp. 120–127. Springer (2010)
13. Murphy, A.H.: A new vector partition of the probability score. Journal of Applied Meteorology 12(4), 595–600 (1973)
14. Nouretdinov, I., Melluish, T., Vovk, V.: Ridge regression confidence machine. In: Proceedings of the 18th International Conference on Machine Learning (ICML'01). pp. 385–392. Morgan Kaufmann, San Francisco, CA (2001)
15. Papadopoulos, H.: Inductive Conformal Prediction: Theory and application to neural networks. In: Fritzsche, P. (ed.) Tools in Artificial Intelligence, chap. 18, pp. 315–330. InTech, Vienna, Austria (2008), http://www.intechopen.com/download/pdf/pdfs_id/5294
16. Papadopoulos, H., Gammerman, A., Vovk, V.: Reliable diagnosis of acute abdominal pain with conformal prediction. Engineering Intelligent Systems 17(2-3), 115–126 (2009)
17. Papadopoulos, H., Haralambous, H.: Reliable prediction intervals with regression neural networks. Neural Networks (2011), http://dx.doi.org/10.1016/j.neunet.2011.05.008
18. Papadopoulos, H., Proedrou, K., Vovk, V., Gammerman, A.: Inductive confidence machines for regression. In: Proceedings of the 13th European Conference on Machine Learning (ECML'02). LNCS, vol. 2430, pp. 345–356. Springer (2002)
19. Papadopoulos, H., Vovk, V., Gammerman, A.: Regression conformal prediction with nearest neighbours. Journal of Artificial Intelligence Research 40, 815–840 (2011), http://dx.doi.org/10.1613/jair.3198
20. Pattichis, C.S., Christodoulou, C., Kyriacou, E., Pattichis, M.S.: Artificial neural networks in medical imaging systems. In: Proceedings of the 1st MEDINF International Conference on Medical Informatics and Engineering. pp. 83–91 (2003)
21. Proedrou, K., Nouretdinov, I., Vovk, V., Gammerman, A.: Transductive confidence machines for pattern recognition. In: Proceedings of the 13th European Conference on Machine Learning (ECML'02). LNCS, vol. 2430, pp. 381–390. Springer (2002)
22. Saunders, C., Gammerman, A., Vovk, V.: Transduction with confidence and credibility. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence. vol. 2, pp. 722–726. Morgan Kaufmann, Los Altos, CA (1999)
23. Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., Johannes, R.S.: Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In: Proceedings of the Annual Symposium on Computer Applications and Medical Care. pp. 261–265. IEEE Computer Society Press (1988)
24. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer, New York (2005)