On Improving Breast Cancer Prediction Using GRNN-CP

Kefaya Qaddoum

***Abstract*—**The aim of this study is to predict breast cancer and to construct a supportive model that will stimulate a more reliable prediction as a factor that is fundamental for public health. In this study, we utilize general regression neural networks (GRNN) to replace the normal predictions with prediction periods to achieve a reasonable percentage of confidence. The mechanism employed here utilises a machine learning system called conformal prediction (CP), in order to assign consistent confidence measures to predictions, which are combined with GRNN. We apply the resulting algorithm to the problem of breast cancer diagnosis. The results show that the prediction constructed by this method is reasonable and could be useful in practice.

Keywords—Neural network, conformal prediction, cancer classification, regression.

I. Introduction

CP is an original method, which is able to complement the predictions of conventional machine learning algorithms by measuring their confidence [4] in order to help to determine how accurate the prediction is, and to consequently suggest good decision-making process. References [4] and [5] proposed ICP to solve the computational ineffectiveness problem of CP.

This work uses a regression CP built on neural networks (NNs). An adjusted CP was needed so as to apply CP to NNs, which is called generalized regression neural network conformal prediction (GRNN-CP). In the case of regression, CPs gives a sufficient level of confidence comparing to conventional techniques.

We used the Wisconsin Breast Cancer Diagnosis (WBCD) dataset [7], which is popular in this domain [3]. We conduct experiments on the dataset and provide results that demonstrate the accuracy of our predictor and the usefulness of the confidence measures. The WBCD dataset was documented at the University of Wisconsin Hospital and contains features which are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass [8]. The cases can be classified as good as indicated in [6].

II. Producing Confidence Information

Machine learning may be used to produce an accepted confidence of information, e.g. the Bayesian framework and ‘probably approximately correct’ (PAC theory) [7], [8]. This experiment will focus on the robustness of prediction intervals for a future independent observation to consider the problem of constructing prediction intervals in a regression state. An advantage of a prediction interval over a point estimate is that it takes into account the variation of the future observation around the point estimate.

An expected failure might occur for the confidence levels to attribute the percentage of expected errors. The next section explains the framework then investigates, via a simulation study, the performance of these prediction intervals in terms of the prediction intervals robustness and their possible uses.

III. The CP Framework

In this section, we describe the idea behind CP, and a more detailed description is provided by [1]. The interest here is in predicting the label of an example $xl{+}g$, based on a set of training examples $\{(x1, y1), . . . , (xl , yl)\}$, where each $xi \in \mathbb{R}_d$ is the vector of attributes: for example, i and $yi \in R$ is the label of that example. The only assumption made is that all (xi, yi), $i = 1, 2, . . .,n$ have been produced from the probability distribution. Main aim of CP [6], [2] is to presume that each probable label $\hat{}y$ is presented in the form of the example $xl{+}g$, to check the possibility to generate the prediction rule:

$$\{(x1, y1), . . . , (xl, yl), (xl{+}g,\ \hat{}y)\} \tag{1}$$

This rule maps every input pattern xi to a predicted label yi:

$$D\{(x1, y1),...,(xl, yl),(xl{+}g,\ \hat{}y)\} \tag{2}$$

The nonconformity total of each pair (xi, yi): $y = 1, . . . , l, l + g$ is then measured as the degree of contention between the prediction and the actual label yi ; it may be noted that, in the case of the pair $(xl{+}g, y)$, the actual label is replaced by the assumed label y. The function used for measuring this degree of contention is referred to as the nonconformity measure of the CP. A change in the assumed label ˆy affects all predictions. Following this, the nonconformity score x$l{+}g$ is compared to the nonconformity results of remaining examples to ascertain how rare the pair $(xl{+}g,\ y)$ is, regarding the nonconformity measure used by the following function:

$$\hat{}yi= D\{x1 , y1), . . . , (xl , yl), (xl{+}g ,\ \hat{}y)\}) \tag{3}$$

The main weakness of the prime CP technique is that, in view of its inspirational quality, all its computations require repeating each new test example for every assumed label. This makes it computationally incompetent. CP is closely efficient [6], and may be merged with any traditional regression technique.

Kefaya Qaddoum is with the Higher Colleges of Technology, United Arab Emirates (e-mail: k.s.qaddoum@gmail.com).

CP splits the training set (of size *l*) into two smaller sets; the convenient training set with $m < l$ examples, and the calibration set with $q := l - m$ examples. Then, it uses the convenient training set for training, and the calibration set for calculating the probability distribution of each possible label *y* for *(x1, y1), . . . , (xm, ym)* to generate the prediction rule, where the nonconformity of each example in the calibration set is *i= 1, ..., q*, and the confidence level will be calculated as *1- δ* which will provide the minimum and maximum *ˆy.*

CP algorithm requires a key parameter that is the number *q*, of training examples to be allocated to the calibration set, while the nonconformity scores will be used by the CP to create its prediction intervals. This number is critical and should only relate to a small portion of the training set, where removing these examples causes a significant decrease in the predictive capability of the NN, and accordingly to broader prediction intervals.

IV. GRNN-CP Framework

The GRNN created by [10] is an estimation method for function regression that has been applied to engineering and science applications. GRNN is useful since it could employ few training samples to converge to the underlying function of the data available. GRNN also is a useful tool to perform predictions and comparisons of system performance in practice. The standard GRNN in Fig. 1 can be implemented with a rapid training procedure due to the single training parameter σ. Finally, it does not require an exact topology definition such as the MLP, or basis function centres and weights, such as the RBF.

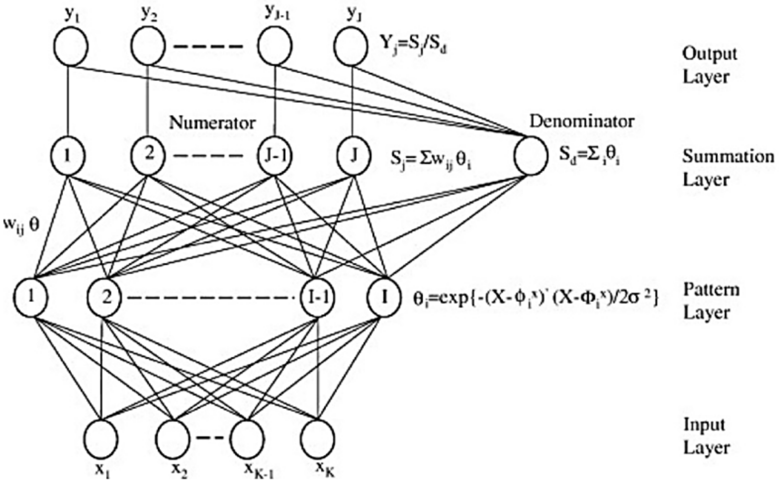

Fig. 1 General GRNN architecture (adopted from [10])

Utilising CP with GRNN has the advantage of enabling much better control of the smoothness of the approximation so that the regression surface adapts to the local properties of the data. In order to use CP in conjunction with some traditional algorithms, a nonconformity measure first needs to be defined. As previously discussed, a nonconformity measure is a function measuring the contention between the actual label *yi* and prediction *yi*' produced by the prediction rule described by [6] of the underlying algorithm for the example *xi*. Regression, meanwhile, can be readily defined as the absolute difference between the two. This section describes the GRNN in CP (GRNN CP) algorithm, and defines a normalized nonconformity measure, which has the effect of producing tighter prediction intervals by taking into account the expected accuracy of GRNN.

The GRNN predicts continuous outputs. GRNN nodes require two main functions to calculate the difference between all sets of input pattern vectors and estimate the probability density function of the input variables. Euclidean distance is used to calculate the difference between input vectors between data values in attribute space. Weighting the calculated distance of any point by the probability of other points occurring in that area concedes a predicted output value shown in (4)

$$\mathrm{D}(\mathrm{x}, \mathrm{x_i}) = \sum_{\mathrm{j}=1}^{\mathrm{p}} \left(\mathrm{x_j} - \mathrm{x_{ij}}/\sigma_{\mathrm{j}}\right)^2 \tag{4}$$

where yi is the ith case actual output value, D(x, xi) is calculated from (5), and n is the total number of cases in the dataset.

$$\mathrm{D}(\mathrm{x}, \mathrm{x_i}) = \sum_{\mathrm{j}=1}^{\mathrm{p}} \left(\mathrm{x_j} - \mathrm{x_{ij}}/\sigma_{\mathrm{j}}\right)^2 \tag{5}$$

where x is the input vector, xi is the ith case vector, xj is the jth data value in the input vector, xij is the jth data value in the ith case vector, and σj is the smoothing factor (Parzen's window) for the jth variable [7]. The error is measured by the means of the mean square error (MSE). The MSE measures the average of the square of the amount by which the estimator differs from the quantity to be estimated. While finding the error, the calculation mentioned earlier will be running frequently with different smoothing factors (sigmas) [9]. Training stops when either a threshold minimum square error value is reached or the test set square error concluded. Since the aim is to produce a level of confidence information, we employ GRNN here to complement predictions with probabilistic texture. The purpose of the global parameter σ is to regulate smoothness of the regression surface. However, as discussed previously, because the data density can vary in different regions, different values of σ may be needed for different patterns *xi*. Allocating an individual σ i for each i^{th} pattern in (5) and combining with (6) produces the standard GRNN as follows:

The smoothness parameter was arbitrarily chosen to σ =0.1. As explained earlier in section 3, CP splits the training set *{(x1, y1), . . . , (xl, yl)}* into two subsets: the convenient training set: *{(x1, y1), . . . , (xm, ym)}*, and the calibration set: *{(xm+1, ym+1), . . . , (xm+q, ym+q)}*.

$$\alpha i = |yi - \hat{}yi| \qquad (7)$$

The GRNN-CP continues as follows:

Sort the nonconformity scores in descending order achieving the following order

$$\alpha(m+1), . . . , \alpha(m+q) \qquad (8)$$

For each new test example *xl+g*: supply the input pattern *xl+g* to the trained GRNN to get the prediction *ˆyl+g* and output the prediction interval

$$(\hat{}yl+g - \alpha(m+s), \hat{}yl+g + \alpha(m+s)) \qquad (9)$$

where s = δ(q + 1).

V. Experimental Evaluation

The proposed method has been tested on the WBCD dataset which contains 683 instances with nine integer valued attributes for each instance. Prior to conducting the experiments in this section, datasets were normalized to the range between [-1, 1]. A random split has been conducted into k folds, and the trials were repeated k epochs, each using one k fold to be tested, and the other k − 1 folds to be the training set. The Campari and Cherry datasets were split into 10 and 4 folds, correspondingly. The calibration set size q was approximately 1/10th of each training portion of the dataset's size.

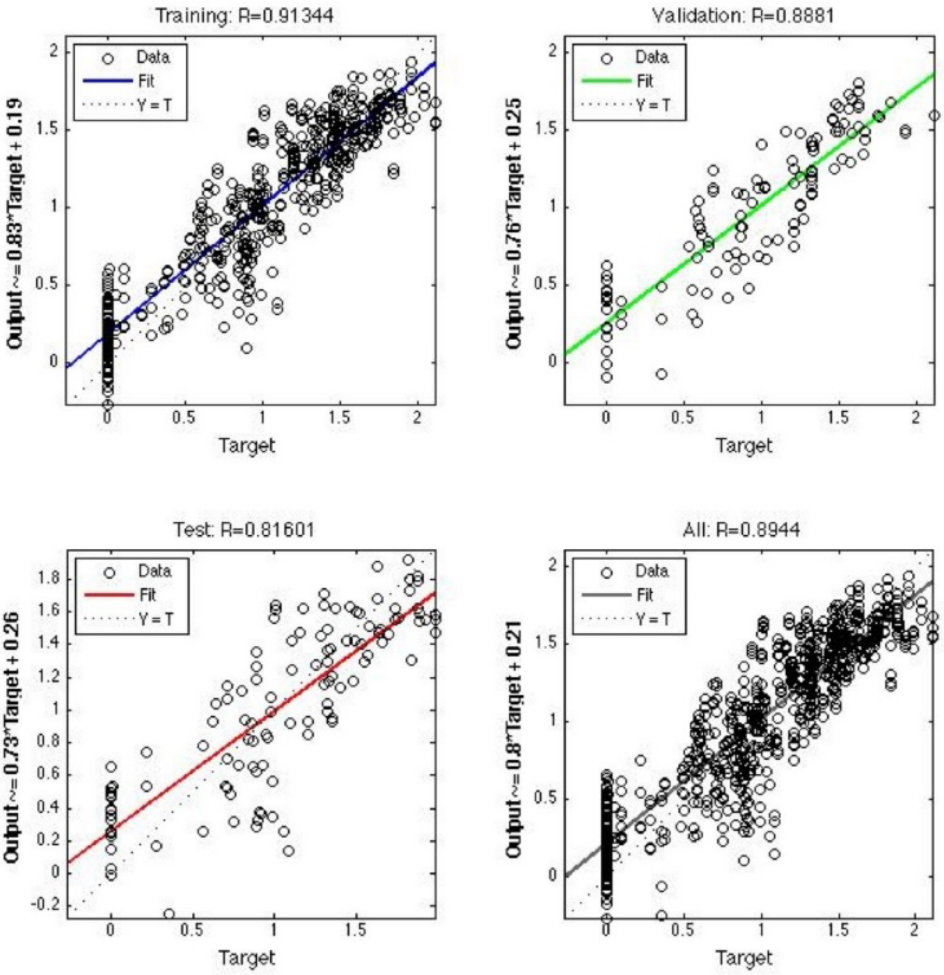

Fig. 2 Prediction intervals regression without GRNN-CP

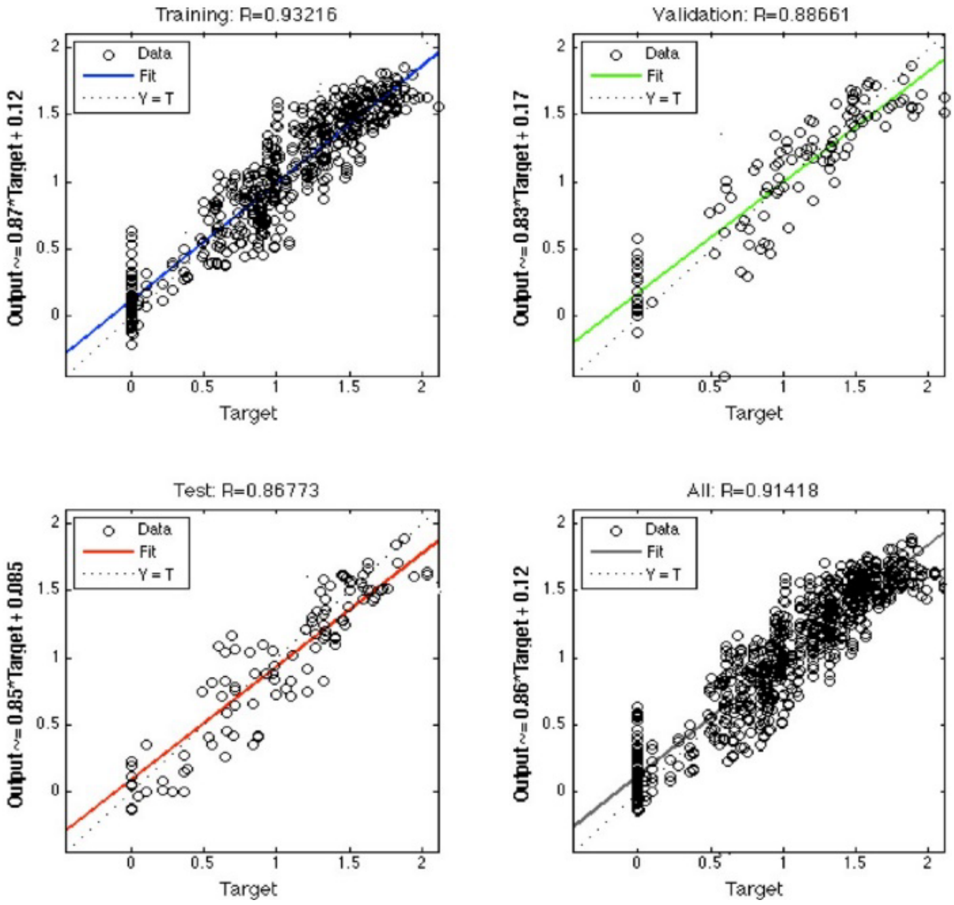

Fig. 3 Prediction intervals regression with GRNN-CP

Trial and error determined the number of hidden neurons, through a fold cross-validation process, with the GRNN predictor on random combinations, which were different from those that evaluated the GRNN- CP. The training algorithm stopping condition is based on a validation 10% of training examples. The GRNN was applied to both calibration and test samples.

$$RMSE = \sqrt{\frac{1}{N}\sum(OBSERVED^2 - PREDICTED^2)} \qquad (10)$$

The performance of the point predictions of the method used in this section, comparing its predictions to the measured values can assess a model trained on the training set. These values are determined by frequently modified various model parameters. The performances of the models are evaluated in terms of its root mean squared error (RMSE), see Figs. 2 and 3.

Table I shows the degree of certainty and the error rates of the CP with non-conformity measures, given four confidence levels: 99%; 98%; 95%; and 90%. The certainty is measured in terms of how many "predictive regions" contain only a single prediction (i.e. only a single p-value is above a given significance level for the new instance we wish to predict). We do this, in order to test the efficiency of our confidence measures, since we would like to have as many certain predictions as possible, given high confidence levels.

The resulted values show that the prediction intervals produced by the method developed in this chapter are quite tight. The median widths obtained using nonconformity measures are 79.4% and 51.2% of the label range of the two datasets correspondingly, while the best widths achieved using the nonconformity degree are 66.5%, and 37.3% of the label range.

TABLE I
CONFIDENCE, ERROR RATES, AND CERTAINTY MEASURES

Confidence Level	Certainty	Error
96%	83.9%	0.7%
98%	94.0%	1.7%
95%	99.5%	4.8%
90%	98.7%	6.8%
99%	82.4%	1.1%
98%	94.8%	1.7%

VI. CONCLUSION

A new prediction system has been constructed in this paper. The proposed algorithm is based on using CP to find the most reliable prediction regressions using GRNN, in order to achieve low errors and more reliable predictions as the results show. The tests performed on the proposed training algorithm show that a good level of accuracy may be achieved when compared to other models.

A relatively good correlation was observed between the measured and predicted values using hybrid GRNN-CP method. The proposed algorithm produces prediction intervals to fulfill a suitable confidence level. In terms of point predictions, performed correlation coefficient between the predicted and the actual values was convenient; For example, 92% confidence level covers 21.5% of the data, while for the 94% confidence level it covers 13.7%. It is worth mentioning that the prediction intervals produced by the proposed method are not only well calibrated, and therefore highly reliable, but they are also tight enough to be useful in practice. In addition, GRNN-CP made progress in terms of prediction interval tightness over the typical regression measure, but it still could be improved by reaching more tightness when it comes to prediction regression, also other regression techniques could be implemented and tested with CP, taking into consideration that adding extended datasets with more records could enhance prediction confidence.

References

[1] H. Papadopoulos, "Regression conformal prediction with nearest neighbours," *J. Artificial Intelligence Res.*, vol. 40, pp.815–840, Jan 2011.

[2] D. F. Specht, "A general regression neural network," IEEE Trans. Neural Netw., vol. 2, no. 6, pp. 568–576, Nov. 1991.

[3] Umesh D R, B Ramachandra, "Association rule mining based predicting breast cancer recurrence on SEER breast cancer data", Emerging Research in Electronics Computer Science and Technology (ICERECT) 2015 International Conference on, pp. 376-380, 2015.

[4] H. Holst, M. Ohlsson, C. Peterson, L. Edenbrandt, "Intelligent computer reporting 'lack of experience': a confidence measure for decision support systems", *Clinical Physiology*, pp. 139-147, 1998.

[5] H. Papadopoulos, A. Gammerman, V. Vovk, "Confidence Predictions for the Diagnosis of Acute Abdominal Pain", *Proc. 5th IFIP Conference on Artificial Intelligence Applications & Innovations*, pp. 175-184, 2009.

[6] A. Gammerman, I. Nouretdinov, B. Burford, A. Chervonenkis, V. Vovk, Z. Luo, "Clinical mass spectrometry proteomic diagnosis by conformal predictors", *Statistical applications in genetics and molecular biology*, vol. 7, no. 2, 2008.

[7] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, Irvine, CA:University of California, School of Information and Computer Science.

[8] O. L. Mangasarian, W. H. Wolberg, "Cancer diagnosis via linear programming", *SIAM News*, vol. 23, no. 5, pp. 1-18, 1990.

[9] Kefaya Qaddoum, Evor Hines, and Daciana Iliescu, yield prediction technique using hybrid adaptive neural genetic network, Int. J. Comp. Intel. Appl. 11, 1250021 (2012) (15 pages) DOI: http://dx.doi.org/10.1142/S1469026812500216.

[10] A general regression neural network, D. F. Specht, October 1991 IEEE Transactions on Neural Networks: Volume 2 Issue 6, November 1991.