Al-Nima RRO, Dlay SS, Woo WL, Chambers JA.

A novel biometric approach to generate ROC curve from the Probabilistic Neural Network.

In: 24th Signal Processing and Communication Application Conference (SIU) 2016. 2016, Zonguldak, Turkey: Institute of Electrical and Electronics Engineers.

DOI link to article:

http://dx.doi.org/10.1109/SIU.2016.7495697

Date deposited:

09/12/2016

A Novel Biometric Approach To Generate ROC Curve From The Probabilistic Neural Network

R.R.O. Al-Nima, S.S. Dlay, W.L. Woo and J.A. Chambers
ComS^{2}IP Research Group, School of Electrical and Electronic Engineering, Newcastle University, England, UK
r.r.o.al-nima@ncl.ac.uk, satnam.dlay@ncl.ac.uk, Lok.Woo@ncl.ac.uk, Jonathon.Chambers@ncl.ac.uk

***Abstract*—The aim of this paper is to present a new method to produce a Receiver Operating Characteristic (ROC) curve from a Probabilistic Neural Network (PNN). Traditionally, an ROC curve has been used widely to report the recognition system measurements. Two main problems arise when using the PNN. Firstly, the PNN outputs are always logical (zeros and one); secondly, a PNN is considered as a multi-class classifier, because it usually has more than one output class. To solve these problems, we suggest a new approach to acquire the score values from the PNN, establish the relationship between the ROC parameters for each class and fusing them to generate one main ROC curve. Personal authentication based on the Finger Texture (FT) biometric has been used to collect the ROC parameters, where three feature extraction methods have been implemented and evaluated: Coefficient of Variance (CV) statistics, Gabor filter followed by the CV calculations and Local Binary Pattern (LBP) followed by the CVs. The results show the accuracy of the Equal Error Rates (EERs) recorded for each ROC graph compared with the actual practical values.**

***Index Terms*—Biometric, finger texture, probabilistic neural network, ROC curve**

I. Introduction

An ROC graph is a measuring method used widely to evaluate verification or identification systems. It consists of different parameters, the False Acceptance Rate (FAR), False Rejection Rate (FRR) and True Positive Rate (TPR) [1]. Furthermore, the trade off point which is considered as an essential parameter to evaluate any recognition system is the EER. Basically, if the EER has a small value this means that the system is efficient and if the EER has a large value this reports that the recognition is inefficient.

A relationship will be established between the FAR and TPR (mathematically this equals to 1-FRR) for each classifier [1] according to an adaptive threshold. FAR represents a matching value which is greater than the threshold and FRR represents a matching value which is less than the threshold [2]. That is, at each threshold the percentage of the correctly classified genuines in a recognition process will be considered as TPR and the percentage of the incorrectly classified impostors will be recorded as FAR [1].

A. Related Work

To start with, a new strategy to produce an ROC for an Artificial Neural Network (ANN) is proposed in [1]. The main idea is to utilize the bias in the hidden layer to give positive and negative offsets and calculate the ANN outputs as scores. The main problem in [1] is that their suggestion is just for a Multi-Layer Perceptron (MLP) neural network with a bias and for two classes only rather than multi-class. Commercial software was used in [3] to draw the ROC graph for estimating mortality risk with a PNN, but the curve was not smooth and it was designed for a PNN with only two classes. In [4] an investigation was strengthened as the results were averaged after running the PNN many times. In addition, a combination method to establish the ROC has been used in [5], where a Parzen PNN (PPNN) and a Gabor filter were employed for periocular recognition. Similarly, a PPNN and kernel discriminant analysis were used together to produce the ROC for the iris classification purpose [6]. Furthermore, in [7] a combination of multiple neural networks was used to collect the ROC parameters, where a voting process was applied for the decision of three neural networks. Obviously, this will increase the complexity of the system.

It is worth mentioning that many publications employed the PNN in the case of biometric recognition without constructing the ROC curve such as [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18]. It is clear that there is no specific method to establish the ROC graph from the multi-classifier PNN unless employing a combination process with other techniques, which will cause more complexity and be time consuming. Table I provides a summary of related publications reported in this literature.

TABLE I: Summary of related approaches for generating the ROC graph from a PNN

Reference	Existing ROC ?	Problem of ROC curve
[1]	Yes	Utilizing the bias of the hidden layer. It is just for an MLP with a bias and only two classes
[3]	Yes	Using commercial software for PNN with only two classes. The curve was not smooth
[4]	Yes	The results were averaged after running the PNN many times
[5]	Yes	A combination method using a PPNN and a Gabor filter
[6]	Yes	A combination method using a PPNN and kernel discriminant analysis
[7]	Yes	A combination of multiple neural networks then applying a voting process
[8],[9],[10],[11],[12],[13],[14],[15],[16],[17] and [18]	No	—

B. Research Aim and Paper Structure

The aim of this contribution is to present a new approach to generate the ROC graph from a multi-class PNN.

The rest of this paper is organized as follows. Section II describes the PNN structure. Section III illustrates how to produce the ROC from the PNN. Section IV explains the proposed practical experiments on the FTs with the results and comparisons. Section V provides the conclusions.

II. PNN

A PNN is a supervised neural network. It consists of multiple layers: the input layer, the hidden or pattern layer, the summation

978-1-5090-1679-2/16/$31.00 2016 IEEE

layer and the decision or output layer, respectively [19]. Fig. 1 shows the main structure of the PNN.

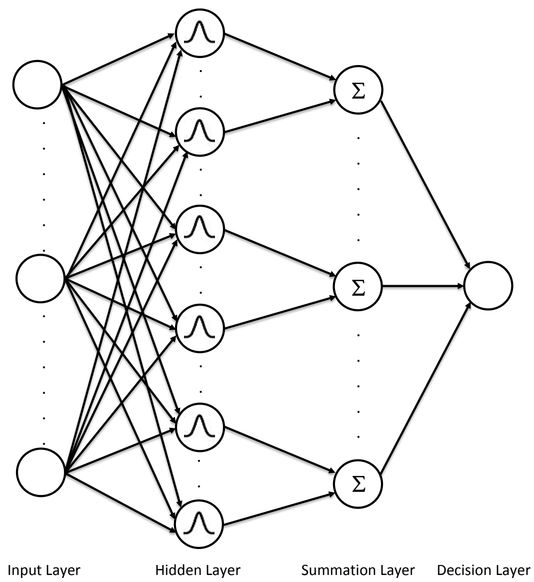


Fig. 1: The main structure of the PNN

The input layer merely sends the input vector to the hidden layer through the first connection weights. Radial Basis Functions (RBFs) are used in the second layer (the hidden layer) to calculate the distance between the input vector and the weights. In the summation layer a summation operation is performed for the hidden output values for each class and their connection weights are created immediately from the targets, where the connection weight will equal to one if the input pattern belongs to the specific class and equal to zero otherwise. Finally, a decision output will be given according to the competitive rule (winner takes all).

To calculate the performance of the PNN, the following equation is used in the pattern layer [19]:

$$\mathbf{z} = exp\left[-\frac{(\mathbf{x}-\mathbf{w}_i)^T(\mathbf{x}-\mathbf{w}_i)}{2\sigma^2}\right] \quad , \quad i = 1,2,...,p \tag{1}$$

where, $\mathbf{z}$ is the output vector of a hidden or pattern nodes, $\mathbf{x}$ is the input vector $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ and $\mathbf{w}_i$ is a training vector $\mathbf{w}_i = [w_1, w_2, ..., w_n]^T$, and $(.)^T$ denotes vector transpose.

Consequently, the summation layer will give the exact probabilistic values of each class for the same input vector. Hence, the decision layer picks the maximum of the summation values S_j and provides the target class $Tclass$ for the input vector as shown below [10]:

$$Tclass(x) = argmax\{\mathbf{s}_j(x)\} \quad , \quad j = 1,2,...,c \tag{2}$$

As for other neural networks, the PNN requires two stages: training and testing stage. The main advantage of this network is that it is very fast during the training stage, because it does not iterate to establish the matching weights between the inputs and their targets [3] [20].

During the training stage, the connection weights between the input and the hidden layer are set equal to the specific training samples. Whereas during the testing stage, the hidden layer computes the distance from the input vector to the training patterns and generates a vector whose input is close to the training pattern [10]. The summation layer contributes in producing vector elements, where each element describes the overall probability of the input vector to each class. These summation values will be observed, any summation node achieving the maximum value will be considered as a competition winner. In the decision layer, the class winner will be represented as '1' while all other classes will be set to '0's and there will always be a winner [20].

From this point, it can be noticed that the summation layer holds the actual output values, which can be considered as score values to produce the ROC curve. On the other hand, the output layer is just a logical decision of the corresponding values in the summation layer.

III. Generation of ROC for PNN

Generally, ROC parameters are calculated for each single class of a neural network. Assuming an MLP network has one output node to classify two classes, O represents the neural network output values and T represents the target with one and zero. A set of $\{O,T\}$ will be arranged and an adaptive threshold is applied through this set [1] [21]. In other words, the conventional method for establishing the ROC curve is by varying a threshold through the output node, where the output values are considered as scores and the relationship is constructed with its targets.

ROC parameters for each class can be computed according to the following outcomes:

- True Positive (TP), if the neural output has correctly classified the positive case.
- False Positive (FP), if the neural output has incorrectly classified the positive case.
- True Negative (TN), if the neural output has correctly classified the negative case.
- False Positive (FN), if the neural output has incorrectly classified the negative case.

Thus, a confusion matrix is found as shown in Table II.

TABLE II: The confusion matrix

TP	FP
FN	TN

Moreover, the following equations will be computed and collected for each thresholding instance [21]:

$$TPR = \frac{Positives\ Correctly\ Classified}{Total\ Positives} \tag{3}$$

$$FAR = \frac{Negatives\ Incorrectly\ Classified}{Total\ Negatives} \tag{4}$$

The main challenge in producing the ROC curve is how to get the score values, which are one of the main parameters of the $\{O,T\}$ set in this matter. It could be argued that the actual output is in the summation node of the jth class which can lead to the key idea of this paper. However, after analysing these values we can see that they require a remapping process. That is because, according to Equation (2) some very small values could win the competition. So, to address this problem a relationship between the outputs of the summation layer and the target class has been established and implemented according to the following equation:

$$PNN_{Score_j} = \begin{cases} PNN_{Score_j} \times Fac1 & \text{if} \quad Tclass_j = 1 \\ PNN_{Score_j} \times Fac2 & \text{if} \quad Tclass_j = 0 \end{cases} \tag{5}$$

where, j is a counter of the summation or decision nodes $j = (1, 2, ..., c)$, $Tclass$ is the desired target, PNN_{Score} is the output of the summation nodes, $Fac1$ and $Fac2$ are scaling factors and $Fac2$ can be denoted as $Fac2 = 1/Fac1$.

After collecting the ROC parameters (TPR) and (FAR) it is easy to draw the ROC curve as there is commonly a relationship between them. To combine this process for all classes in a multi-class PNN, average processes can be performed for all TPRs and FARs. This will lead to the generation of the main smoothing ROC curve that describes the PNN performance.

IV. Experimental Context

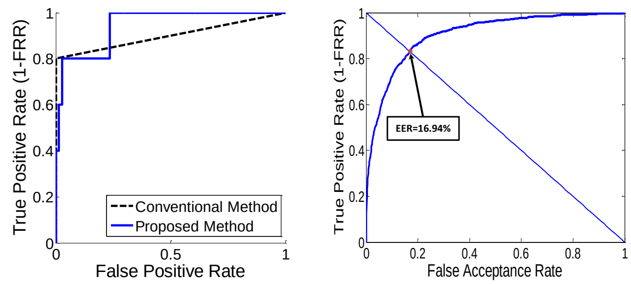
Our proposed method is evaluated using biometric authentication based on the human FTs. Due to the fact that, it is more important to concentrate on getting different results to evaluate the ROC performance, different feature extractions have been employed for four finger images (index, middle, ring and little). These images have been acquired from the Hong Kong Polytechnic University Contact-free 2D Hand Images Database version 1.0 [22]. Then, the Region of Interest (ROI) of each finger image has been specified by using a similar suggested method in [23]. Three feature extractions have been applied to collect the finger textures: simple statistics using the Coefficient of Variance (CV), Gabor filter followed by the CV calculations and LBP followed by the CV calculations. For more information about the CV, Gabor filter and LBP can be found in [24], [25] and [26] respectively. In the case of normalization, all of these methods will be passed through the same processing, that is, the same ROI resize before the feature extraction and the same image segmentation and preparation. Each input image is segmented into non-overlapped matrices with fixed sizes 5×5. The image resize is determined equal to 40×170 for all ROIs. Same operations about calculating the CVs are repeated after the Gabor filter and LBP images.

The advantages of using the CV are: to reduce the input vector size, the variances between the features for the same subject are well described, the variances between the features for the different subjects are well described, no dimension units can be considered and all values are small and positive.

A. Results

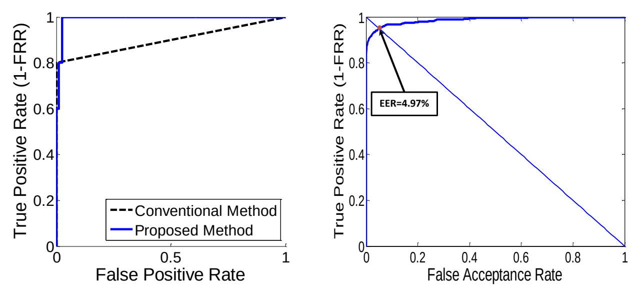
A database from the Hong Kong Polytechnic University Contact-free 2D Hand Images Database version 1.0 has been used in this paper. It consists of 1770 right hand images acquired from 177 subjects in a contact-free manner with a black background. The participants were mainly students and staff from both genders (male and female), but with different ages (from 18 to 50 years). A commercial available 3D digitizer, Minolta VIVID 910 is used to capture 3D and corresponding 2D hand images will a resolution of 640×480 pixels. Each subject provides 10 images in two sessions. The time lapse between them is varied between (1 week - 3 months). Each hand was held at a distance of about 0.7 m from the scanner and the participants were ordered to remove any jewellery. An indoor environment was provided to capture the hand images [22].

In this approach, 5 samples have been used for the training and the same number for the testing. The number of output values in PNN summation and decision layer are the same. They were equal to 177 and this is exactly equal to the number of people who provided their image database. However the output of the decision layer is a logical operation as it always holds '1' for the winner and '0's in all other nodes and there should always be a winner. So, the number of the failed samples has been counted and recorded for each feature extraction method. Consequently, the ROC parameters (FAR and TPR) were collected for each class. A clear demonstration of ROC curves for a single class PNN, which has been selected randomly, are given in Figs. 2a, 3a and 4a. These figures show the differences between the ROC curves of our proposed method versus the conventional method.



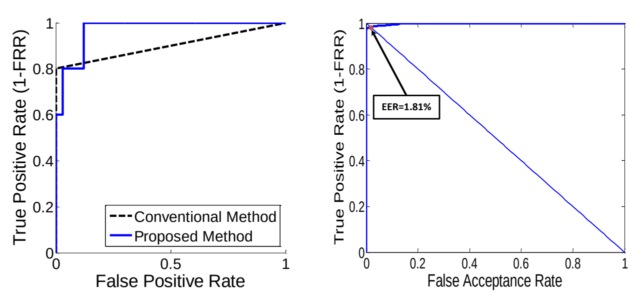
(a) (b)

Fig. 2: ROC curves for single and multi-class PNN:
(a) ROC curves for a single class PNN of the CV method (class No. 142)
(b) The final ROC curve for the multi-class PNN of the CV method



(a) (b)

Fig. 3: ROC curves for single and multi-class PNN:
(a) ROC curves for a single class PNN of the Gabor filter+CV method (class No. 120)
(b) The final ROC curve for the multi-class PNN of the Gabor filter+CV method



(a) (b)

Fig. 4: ROC curves for single and multi-class PNN:
(a) ROC curves for a single class PNN of the LBP+CV method (class No. 169)
(b) The final ROC curve for the multi-class PNN of the LBP+CV method

Hence, to generate the final ROC curve for the multi-class PNN, two average processes have been implemented for all classes. One for the FAR values and another one for the TPR values. This will combine the ROC parameters for all classes together and produce one smoothing ROC curve referring to the total PNN performance. In other words, the relationships between the main ROC parameters have been fused together for all classes and one essential ROC curve was generated. See Figs. 2b, 3b and 4b for the ROCs of the three suggested feature extraction methods with their recorded EERs specified by using the graph.

Moreover, Table III shows the comparisons between the different methods. As it can be seen the EER values in Table III are so close to the EER values determined from the Figs. 2b, 3b and 4b.

TABLE III: Methodology and Performance Comparisons

Reference	Methodology	Failed samples	EER	AUC
Proposed Methods	LBP+CV	16	1.81%	0.998
	Gabor+CV	44	4.97%	0.987
	CV	150	16.95%	0.905
[23]	CompCode	—	6%	0.986

In this table the LBP followed by the CV feature extraction method attained the best results compared with other suggested methods. It seems that this method could be considered as one of the best methods for the FT features.

In addition, the Area Under the Curve (AUC) for the second method (Gabor+CV) is slightly greater than the value of the AUC in [23] as the percentage of the calculated EER is slightly lower than the percentage of the EER in [23]. This is strong evidence in our proposed ROC curve as it could achieve the AUCs and their corresponding EERs even if there is a small difference between the values. Furthermore, Figs. 3b, 4a and 4b show how this suggested approach is able to demonstrate different results accurately.

V. Conclusion

The contribution of this study is to propose a novel method to generate an ROC graph for a PNN. The key idea of this research is to extract the score values from each single class of the PNN. These values can be found in the summation layer of this network, where the summation nodes hold the actual output values. However, these values were remapped or recalculated according to their relationships with the class targets, where the relationship between the summation layer values and the target class has been established and implemented. This method seems more efficient than using the logical output values from the decision layer. After collecting the ROC parameters (FAR and TPR) from each single class, a relationship was constructed to produce the ROC curve. Afterwards, each ROC parameter was fused or combined to all single classes by using the average operation. Therefore, a smooth ROC curve has been produced to represent the multiple classes. A large database and different methods have been used to analyse the work. Generating the ROC graph has never been considered in the multi-class PNN as far as we know and a substantial literature review was undertaken to confirm this fact.

References

[1] K. Woods and K. Bowyer, “Generating ROC curves for artificial neural networks,” *Medical Imaging, IEEE Transactions on*, vol. 16, no. 3, pp. 329–337, 1997.

[2] Ş. Sağiroğlu and N. Özkaya, “An intelligent face features generation system from fingerprints,” *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 17, no. 2, pp. 183–203, 2009.

[3] R. K. Orr, “Use of a probabilistic neural network to estimate the risk of mortality after cardiac surgery,” *Medical Decision Making*, vol. 17, no. 2, pp. 178–185, 1997.

[4] S.-Y. Ooi, A. Teoh, and T.-S. Ong, “Compatibility of biometric strengthening with probabilistic neural network,” in *Biometrics and Security Technologies, 2008. ISBAST 2008. International Symposium on*, pp. 1–6, 2008.

[5] A. Joshi, A. Gangwar, R. Sharma, A. Singh, and Z. Saquib, “Periocular recognition based on Gabor and Parzen PNN,” in *Image Processing (ICIP), 2014 IEEE International Conference on*, pp. 4977–4981, 2014.

[6] R. Sharma, A. Singh, A. Joshi, and A. Gangwar, “Robust iris classification through a combination of kernel discriminant analysis and Parzen based probabilistic neural networks,” in *Emerging Applications of Information Technology (EAIT), 2014 Fourth International Conference of*, pp. 268–272, 2014.

[7] N. Almaadeed, A. Aggoun, and A. Amira, “Speaker identification using multimodal neural networks and wavelet analysis,” *Biometrics, IET*, vol. 4, no. 1, pp. 18–28, 2015.

[8] S.-H. Lin, S.-Y. Kung, and L.-J. Lin, “Face recognition/detection by probabilistic decision-based neural network,” *Neural Networks, IEEE Transactions on*, vol. 8, no. 1, pp. 114–132, 1997.

[9] L. Huang, H. Zhuang, and W. Zhang, “Person recognition using features of still face images and text-independent speeches,” in *Communications, Computers and Signal Processing, 2007. PacRim 2007. IEEE Pacific Rim Conference on*, pp. 284–287, 2007.

[10] S. Meshoul and M. Batouche, “Combining Fisher discriminant analysis and probabilistic neural network for effective on-line signature recognition,” in *Information Sciences Signal Processing and their Applications (ISSPA), 2010 10th International Conference on*, pp. 658–661, 2010.

[11] J. Hossain and M. Amin, “Leaf shape identification based plant biometrics,” in *Computer and Information Technology (ICCIT), 2010 13th International Conference on*, pp. 458–463, 2010.

[12] R. Sundaram and B. Dhara, “Neural network based iris recognition system using Haralick features,” in *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, vol. 3, pp. 19–23, 2011.

[13] S.-J. Chuang, S.-R. Zeng, and Y.-L. Chou, “Neural networks for the recognition of traditional chinese handwriting,” in *Computational Science and Engineering (CSE), 2011 IEEE 14th International Conference on*, pp. 645–648, 2011.

[14] L. Sudha and R. Bhavani, “Biometric authorization system by video analysis of human gait in controlled environments,” in *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, pp. 139–144, 2011.

[15] C. Yu and L. Huang, “Biometric recognition by using audio and visual feature fusion,” in *System Science and Engineering (ICSSE), 2012 International Conference on*, pp. 173–178, 2012.

[16] M. Saini, J. Saini, and S. Sharma, “Moment based wavelet filter design for fingerprint classification,” in *Signal Processing and Communication (ICSC), 2013 International Conference on*, pp. 267–270, 2013.

[17] P. Wibawa, B. Agung, and F. Sthevanie, “Palm print recognition using competitive hand valley detection, local binary pattern and probabilistic neural network,” in *Information Technology Systems and Innovation (ICITSI), 2014 International Conference on*, pp. 105–110, 2014.

[18] D. Perwira, B. Agung, and M. Sulistiyo, “Personal palm vein identification using principal component analysis and probabilistic neural network,” in *Information Technology Systems and Innovation (ICITSI), 2014 International Conference on*, pp. 99–104, 2014.

[19] L. V. Fausett and P. Hall, *Fundamentals of neural networks: architectures, algorithms, and applications*, vol. 40. Prentice-Hall Englewood Cliffs, 1994.

[20] S. Shorrock, A. Yannopoulos, S. Dlay, and D. Atkinson, “Biometric verification of computer users with probabilistic and cascade forward neural networks,” p. 267272, 2000.

[21] T. Fawcett, “An introduction to ROC analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[22] “The Hong Kong Polytechnic University contact-free 3d/2d hand images database version 1.0.”

[23] V. Kanhangad, A. Kumar, and D. Zhang, “A unified framework for contactless hand verification,” *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 3, pp. 1014–1027, 2011.

[24] L. Junli, Y. Gengyun, and Z. Guanghui, “Evaluation of tobacco mixing uniformity based on chemical composition,” in *Control Conference (CCC), 2012 31st Chinese*, pp. 7552–7555, 2012.

[25] I. Fogel and D. Sagi, “Gabor filters as texture discriminator,” *Biological cybernetics*, vol. 61, no. 2, pp. 103–113, 1989.

[26] T. Ojala, M. Pietikäinen, and D. Harwood, “A comparative study of texture measures with classification based on featured distributions,” *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.