Proceedings of the World Congress on Engineering 2007 Vol I
WCE 2007, July 2 - 4, 2007, London, U.K.

Design and Analysis of a novel weightless artificial neural based Multi-Classifier

P. Lorrentz[1], W. G. J. Howells[2], and K. D. McDonald-Maier[3]

Abstract- **Recent years have witnessed intense research in the general area of Multi-Classifier systems (MCS), but this has rarely incorporated any utilisation of weightless neural systems(WNS) as the combiner of an MCS ensemble. This paper explores the application of weightless networks within the multi-classifier environment by introducing an intelligent multi-classifier system using a WNS called the *Enhanced Probabilistic Convergent Neural Networks (EPCN)*. The paper explores the use of EPCN by illustrating its major features, such as the specification of disjoint or overlapping input subset to the MCS, and the inherently parallel nature of the design. Within the proposed system, the number of base classifiers per MCS could be specified manually or automatically. . The proposed MCS is problem-domain independent and, our investigation is performed on handwritten characters. The proposed MCS is adaptive, its combiner is capable of extracting absolute or weighted classification decision(output) from base classifier. Diversity is increased in the base classifier by injecting randomness into the system parameters. Two types of EPCN classifiers are proposed, fix-PCN and rand-PCN. These PCNs are independent and orthogonal. One uses a fixed method of forming connectivity while the other uses random method of forming connectivity.**

In order to verify the performance of the recognition system, tests were performed, off-line, on benchmark datasets of unconstrained handwritten numerals.

Experimental results suggest that MCS outperforms single EPCN in classification of handwritten characters.

Index Term- **Combining strategy, Dynamic reconfiguration, Enhanced Probabilistic Convergent Network, Multi-classifier.**

I. INTRODUCTION

Over recent years, a significant research effort has been devoted to the development of multi-expert systems (MES) and multi-classifier systems (MCS). MES and MCS consist of component classifiers, possibly of an artificial neural configuration, called base classifiers, arranged in a specific fashion so as to carry out a specific task which would otherwise yield a poorer performance should such a task be performed by a single NN or classifier. The specific arrangement of these NNs is commonly referred to as a classifier selection. R. Ranawana [22] summarises various methods used in NN selection but does not significantly include weightless NN. Weighted NN are those NN whose performance and system parameters depend on weights and weight adjustment. In contrast, NNs whose performance and system parameters do not depend on weights (and their adjustments) are called weightless NN. A major advantage of weightless NN is its ease of implementation in digital hardware which makes it highly suitable for implementation in portable embedded systems embedded systems and its ability to efficiently learn with a reduced number of training iteration. In a weightless NN, binary weights are stored and retrieved from RAM. To date, most significant research in NN used in MES and MCS has involve, for example [9] uses classifier selection based on weights. This paper presents an MCS employing weightless classifiers. The base classifiers employed in this work were derived from *The Enhanced Probabilistic Convergent Network, EPCN.* Details of the base classifiers were published in [20].

This paper is organised as follows. Sub-section A introduces EPCN and sub-section B is an introduction to MCS and MES in general. The core investigation commences in section II where an MCS implemented using EPCN is introduced and then tested on unconstrained handwritten characters in section III. The results obtained were recorded and analysed in section IV. The paper completes with the conclusion where it also explores further possible experimentation.

A. *EPCN - The Enhanced Probabilistic Convergent Network*

The architecture of the EPCN comprises primarily the following components, a *pre-group*, and a corresponding *merge-layer* for the *pre-group*, the *main-group*, and *merge-layer* for the *main-group* as depicted in Fig. 2. A feedback path from the *merge-layer* of the *main-group* to the *main-group layers* is included in the design. Each group consist of a number of layers with each constituent layer consisting of component neurons which themselves consist of a number of storage locations known as RAM-locations, as shown in Fig. 1.

[1]Department of Electronics, University of Kent, Canterbury, Kent, CT2 7NT,UK. Tel: 07813089916 E-mail: pl38@kent.ac.uk
[2]Department of Electronics, University of Kent, Canterbury, Kent, CT2 7NT, UK, Tel: +44 1227 764000 x 3389. Fax: +441227 456084, E-mail: W.G.Howells@kent.ac.uk
[3]Department of Computing and Electronic Systems, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK, e-mail: kdm@essex.ac.uk

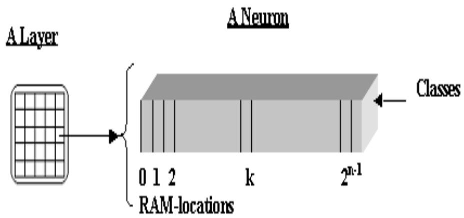

Fig 1: An EPCN neuron.

The EPCN is a classifier which allocates a confidence measure to each candidate class, based on supervised learning, when a pattern is presented to it for classification. An interested reader could consult [20] for more detail.

B. *MCS in general*

MCS consist of component classifiers, possibly of an artificial neural configuration, called base classifiers, arranged in a specific fashion so as to carry out a specific task which would otherwise yield a poorer performance should such a task be performed by a single NN or single classifier.

A significant component within the design process of an MCS system is the selection of the base classifiers to employ. The most common selection methods used for base classifiers are: input data [23], Genetic Algorithm (GA) [4], Objective functions [22], [25], Random selection [26], Boosting [31], and Bagging [25].

Some designers [9], [13] make NN selection to depend on certain diversity measures.

One of the most successful ensemble creation methods is the random subspace method [26]. Here input space is partitioned by random selection into subspaces of equal length and a classifier is assigned to each subspace.

The most common arrangement of base NN used in an MCS is the parallel method. Other topologies are the cascading and hierarchical topology [21]. Cristian Dima [7] proposes the implementation of a hierarchical mixture of experts and the employment of dynamic reconfiguration to analyse robot dynamics.

It is essential that for NN to be included in an MCS, either the performance must be above 50%, or it must make a significant and positive contribution to the ensemble after combination. L. Lam [17] states that

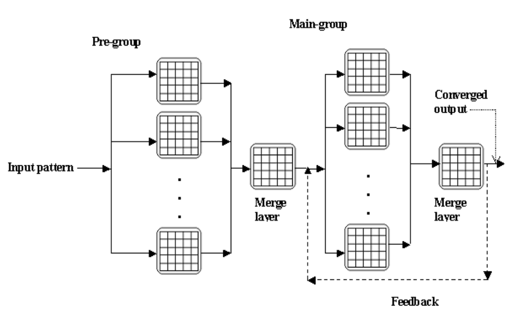

Fig. 2: EPCN architecture.

orthogonality, complementarities and independence of a base classifier determine its inclusion in an MCS. During training and recognition, each base classifier utilises its normal training and recognition algorithm. The combination of base classifier output is called *classifier fusion*. Various techniques for classifier fusion are broadly divided into: *objective functions[25], Qualitative combination [5], Intelligent combiners[21], Fixed combiners or balanced classifiers [11]*. Significantly, EPCN, when used as a combiner, is a novel weightless *intelligent combiner* since it possesses its own learning and recognition algorithms.

A. Krzyzak [18] categorizes combiners of MCS into two, namely, feature-vector-based method (i.e. using neural network) and syntactic-and-structural (i.e. fuzzy-rule based) method. [3] categorises them as: Linear, Non-linear, Statistical, and Computational Intelligent combiners.

The overall performance of an MES or an MCS is often compared to a single base classifier. At present, it is difficult to quantify how diversity measure affect performance, most especially for MCS comprising large number of base classifiers. B. Gabrys and D. Ruta [5] maintain that diversity measure has limited correlation with MCS performance. It should be emphasised that MCS performance depends on careful selection of base classifiers. K. Min [16] use a Rejection criterion and reliability to measure performance. The rejection criterion and reliability are numerical quantities derived from a fuzzy integral. A performance improvement has been made on isolated handwritten characters [6], whole words [15], postal addresses [2], [12], and bank cheques [24]. It is difficult to achieve a high recognition rate using a set of features and a single classifier. This is because totally unconstrained handwritten numerals, as is the case in this work, contain an appreciable level of pattern variation which mainly depends upon individual writing style.

The design of MCS using EPCN as an *intelligent combiner* will be the subject of the next section.

II. THE DESIGN OF MCS FROM EPCN

MCS utilising weightless classifiers are currently rare. This paper presents an MCS that utilises weightless NN called *Enhanced Probabilistic Convergent Network (EPCN)* [20]. MCS may be grouped according to their output. A formal grouping of such classifiers is: abstract form, rank level, and measurement level [22]. Of these, the measurement level group is relevant.

- *Measurement level:-* No attempt is made to arrange the output of a base classier in any order, since the order of values in itself has meaning. Each class is assigned a belief of the classifier about the input. The result is an array of belief values. These classifiers are also called *probabilistic classifiers*. Fix-PCN and rand-PCN are novel weightless *probabilistic classifiers*.

Previous studies have shown the performance of both EPCNs to be well above 50% [20]. Fix-PCN is orthogonal to rand-PCN due to its inherent method of forming connectivity. The rand-PCN uses random method while fix-PCN uses a pre-defined or "fixed" method, a systematic method which is reproducible. These EPCNs are designed to be independent and without correlation with regards to their errors, giving no consideration for any future input dataset. Varying the system parameter of each EPCN has a profound effect on their decision making. These decisions (outputs) do not give rise to error correlation, for disjoint input dataset. Thus EPCNs are good candidates for MCS production.

In this work, the input space is partitioned into overlapping subsets and a classifier is assigned to each subspace. This allows for a clear comparison with a standalone single classifier. Since this MCS uses EPCN, it will henceforth be denoted by MCSPCN for short. MCSPCN is designed with the possibility for dynamic reconfiguration, and the parallel scheme is employed. In a changing environment, system parameters could be made dependent on changes in the environment.

Diversity is increased in MCSPCN by incorporating diversity within the training algorithm [22], [23] of all EPCNs. This influences their behaviour during training and recognition. For example, if F classifier is required, and N_i class each, this will be specified as:

>> mcspcn(F,N_i,r,c); i = 1,2,3, ... -------(1)

where,

r = number of rows in pattern.

c = number of column in pattern.

For each F, the size of N_i may vary or overlap.

This work utilises the *Computational intelligent* method for classifiers fusion by employing another EPCN as the combiner.

A. *Combiner Unit*

The term *Combiner Unit* refers to the EPCN combiner [P_c, M_c], and the gating function *f(.)* (see fig. 3). The gating function consist of a *decision make*r and a *converter.*

Decision maker is required for the following reasons:-

- If the same character is classified or assigned by different NN to differing classes and these classifications are correct, without the *decision maker*, these two interpretation will be converted to different images by the *converter*. A correct classification of a pattern by different NN should produce similar pattern for the EPCN-combiner to train.
- The combiner EPCN does not know if the input space overlaps or not. The *decision maker* is also required to monitor overlap and to reflect this it its output by weighting.

Decision Maker:- The *decision maker* considers the performance of the component classifiers with respect to the classes, and passes its decision to the *converter*. It utilises a weighting strategy on the output of the base classifiers when inputs overlap. This weighting strategy affects only those outputs corresponding to the region of input overlaps. A zero weight switches off an output of a base NN with respect to a given class, while a weight greater than zero switches it on. The decision maker does not eliminate a base classifier, but only inhibits certain outputs with respect to certain classes. This inhibition depends on input space overlap and performance on that class. E.g. if character "a" is trained to one NN as class 1, and trained to another NN as class 2. During recognition, correct classification requires the first NN to classify "a" as class 1 and the second NN should classify it as class 2. The *decision maker* is responsible for telling the converter that the two output are the same i.e. are correct classifications of "a".

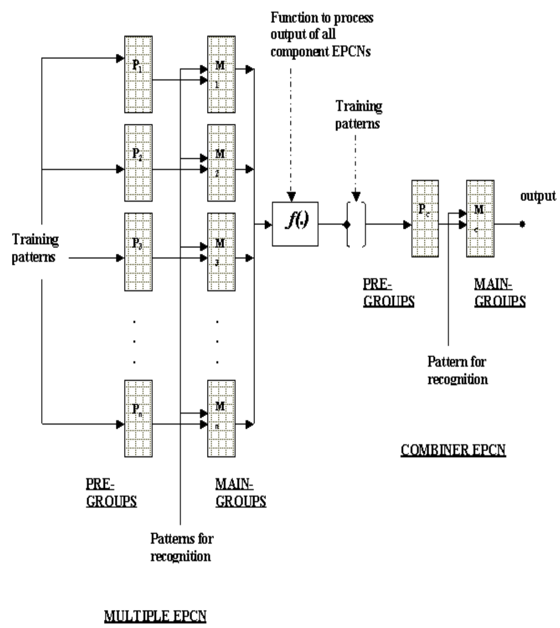

Fig. 3: The MCS unit is divided into multiple EPCN group and combiner EPCN group. The multiple EPCN group consist of EPCNs in parallel. P_i = *pre-group*; M_i = *main-group*; i = 1,2,3,… *f(.)* = gating function. P_c = combiner's *pre-group*. M_c = combiner's *main-group*.

Converter:- This converts the *Decision maker*'s integer output into binary, e.g. for division = 1000, [0, 0, 65, 45, 0, 0] will be converted to:

```
[0000000000
 0000000000
 0001000001
 0000101101
 0000000000
 0000000000];
```

EPCN-combiner configuration:- An example configuration of EPCN combiner is shown below.

```
>> con

con =

    nost: [9 10 8 9 10 10 13 10 6 10]
    nclas: 10
    nlay: 2
    nosnn: 1
    dimz: 10
    dimr: 10
    ntuple: 3
```

In the first field, nost, each number represent the number of training patterns per class. The second field, nclas, represents the total number of classes. The third field is the number of layers in the pre-group. The 4th field is the number of columns in the image while the 5th is the number of rows in the image. The last field, ntuple, is the tuple-size. The combiner's *main-group's* configuration is the same, except the field "nlay" is replaced by "mglay", where "mglay" is the number of layers for the *main-group*. Thus we have a MCS that looks like fig. 3, where $[M_i, P_i]$ is a base classifier; i = 1,2,3,… This multi-classifier will be tested in section III.

III. EXPERIMENTATION

Off-line handwritten characters and numerals recognition has been a topic of intensive research for many years. The performance of EPCN as combiner, should equal or surpass that of feature vector based classifiers or syntactic/structural based classifiers. The MCSPCN is problem-domain independent and as such should perform well on handwritten characters.

The source of totally unconstrained numerals used in this work is:-

- The centre of Excellence for Document Analysis and Recognition (CEDAR), University at Buffalo, State University of New York, Department of Computer Science. Handwritten numbers from CEDAR were resized and binarised to 16-by-24 in dimension.

The pattern used from CEDAR is handwritten numerals "0" to "9". These are thus labelled class 0 to class 9. The number of patterns in each class varies from 200 to 1000 depending on class. These numerals are divided into training patterns and test patterns. Training patterns and test patterns are treated independently. Test patterns do not form part of training patterns and vice versa.

Experiment 1:- The aim of this experiment is to determine if the combiner can successfully interpret result and ignore individual erroneous result from the component classifiers.

In this experiment, the input space was partitioned as shown in Table I. Where, for example, classifier NTW1 is only trained on classes 0 through 4. The component base classifiers, NTW# (where # = 1,2,3,…), are assigned to be trained on the subset of classes depicted in each row of Table I. This sub-setting strategy has been employed in order to artificially lower the performance of each of the base classifiers to observe if the PCN combiner, $[P_c, M_c]$, is able to allow for the poor performances and give a good overall result.

During recognition, each network is required to classify patterns belonging to all the ten classes. All patterns that result were collected in a directory. These were afterwards separated into training set and test set. The training set is used to train the combiner while the test set was employed during recognition. The performance metric used is the percentage (%) of patterns recognised.

Experiment 2:- Obviously in practice, base classifiers would be trained on the entire pattern class set. This second experiment is therefore aimed at determining if the MCS performs better than any of the component classifiers alone.

Experiment 1 is thus repeated with each of the five component classifiers trained on all 10 classes. In practice, this is done by setting F_i =5 and N_i = 10 in MCSPCN (function (1) of section II). During recognition, each network is required to classify patterns belonging to all the ten classes. The results were later collected and processed by the gating function *f(.)*. Again all patterns that result were collected in a directory. These were afterwards separated into training set and test set. Training set is used to train the combiner while the test set was employed during recognition. The performance metric used is the percentage (%) of patterns recognised. All results obtained were processed and important results recorded in Table III.

Table I: Partitioning of the input space in experiment 1; NTW = Base classifier; # = number.

	CLASSES				
NTW1	0	1	2	3	4
NTW2	2	3	4	5	6
NTW3	4	5	6	7	8
NTW4	6	7	8	9	0
NTW5	8	9	0	1	2

Table II: Comparison of a combiner with base neural networks when F = 5; N_i = 5. Clasf. = classifier; NTW# = Network, where # = a number. % = percentage.

Average Performance Comparison (%)			
Component clasf.		Combiner	Difference
NTW1	50	93.37	43.37
NTW2	80	93.37	13.37
NTW3	36	93.37	57.37
NTW4	74	93.37	19.37
NTW5	52	93.37	41.37

Table III: Comparison of a combiner with base neural networks when F = 5; N_i = 10. Clasf. = classifier; NTW# = Network, where # = a number. % = percentage.

Average Performance Comparison (%)			
Component clasf.		Combiner	Difference
NTW1	74	95.14	21.14
NTW2	80	95.14	15.14
NTW3	63	95.14	32.14
NTW4	79	95.14	16.14
NTW5	69	95.14	26.14

IV. RESULT AND ANALYSIS

Table II illustrates the result of experiment 1 and is obtained when F = 5; N_i = 5 is specified to MCSPCN (function (1) of section II) with the databases specified in section III is employed. Table III represents the result of experiment 2 and is obtained when F = 5; N_i = 10 is specified to MCSPCN (function (1) of section II) with the databases specified in section III were employed . Averages were calculated with respect to training set.

The first column of both tables shows the component classifiers, the second column shows their respective performances, and the third column shows the overall performance of the MCSPCN. In table II, NTW1 shows an average (50%) recognition rate while NTW2 shows a high percentage recognition rate (80%). NTW3 shows a poor recognition rate (36%) while NTW4 shows a high percentage recognition rate (74%). In table III, NTW1 shows an average (74%) recognition rate while NTW2 shows a high percentage recognition rate (80%). NTW3 shows a fairly good recognition rate (63%) while NTW4 shows a high percentage recognition rate (79%). From this trend, it could be inferred that when some base classifier performs fairly well on a database, others perform very well on the same database. This shows the inherent orthogonal properties of fix-PCN and rand-PCN.

Comparing the second column of both tables, the classifiers are seen to perform better when trained on all ten classes than when trained on sub-section of the classes. This affects the combiner positively with an average improvement of about 2%.

In table II, the performance of the combiner (at 93.37%) was well above that of the component classifier and shows that the combiner is able to filter out poor component classifier results. In table III, the performance of the combiner (at 95.14%) was also well above that of the component classifiers. From this we may deduce that the gating function *f(.)* considers only the merits of the base classifiers. The individual entries in the difference column (in %) show the performance of the combiner over their corresponding base classifiers.

V. CONCLUSION

In this paper, we have focused on a multi-classifier combining strategy using a novel RAM-based artificial neural network EPCN. The combiner of the multi-classifier has been shown capable of interpreting results from component classifier and ignoring individual erroneous results. Significantly, the multi-classifier is shown to have achieved a high performance rate (93.37% in Table II, and 95.14% in Table III) compared to the component classifiers. It is to be noted also that this performance compares favourably well with other multi-classifiers derived from weighted base classifier or neural network, using other techniques, e.g. [16],[19]. It has been shown that the MCSPCN is problem-domain independent MCS and has performed well on handwritten characters.

Areas of further investigation may include other configuration methods, such as Boosting, Bagging, or using performance criteria to initialise and choose base classifiers. As this is likely to have the effect of eliminating such network as NTW3 (at 36 % in Table II) from the WNS since its performance is sometimes below 50%.

Experimental results suggest that MCS outperformed single EPCN [20] in classification of handwritten characters.

REFERENCES

[1] A. Canuto, M.C. Fairhurst, W.G.J. Howells, Improving Artmap learning through variable vigilance., International Journal of Neural Systems, voln.11 No.6, 509-522., World Scientific Publishing Company. 2001.

[2] A. Kaltenmerier, T. Caesar, J. Gloger, E. Mandler, Sophisticated topology of Hidden Markov Models for Cursive Script Recognition, in: Proc. Of 2nd Int. Conf. On Document Analysis and Recognition, Tsukuba Science City, Japan, pp. 139 – 142, 1993.

[3] A. M. de P. Canuto, Combining Neural Networks and Fuzzy Logic for Application in Character Recognition, PhD Thesis, University of Kent, 2001.

[4] Abhinar Saxena, Ashraf Saad, Evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems, Applied soft computing ASOC – 214, 25 October 2005.

[5] Bogdan Gabrys, Dymitr Ruta, Genetic algorithm in classifier fusion, Applied soft computing 6, pp. 337 – 347, 2006

[6] C. Suen, L. Nadal, R. Legault, T. Mai, L. Lam, Computer Recognition of unconstrained Handwritten Numerals, Proc. Of the IEEE, 80(7), pp. 1162 – 1180, 1998.

[7] Cristian Dima, Sensor and Classifier for outdoor obstacle detection, The Robotic Institute Carnegie Mellon University, 8 April 2003.

[8] D.C. Montgomery et al., (2001), Engineering statistics, John Wiley & Sons Inc., Second Edition, 2001.

[9] Dunja Mladenić, Janez Brank, Marko Grobelnik, Natasa Milic-Frayling, Feature Selection using Linear Classifier Weights: Interaction with classification Models SIGIR '04, Sheffield, South Yorkshire, UK, July 25 – 29 2004.

[10] F. O. Karray, C. de Silva, Soft Computing and Intelligent System design, Addison Wesley, First Publication, 2004.

[11] F. Roli, G. Giacinto, Hybrid Method in pattern recognition, chapter design of multiple classifier systems, Worldwide Scientific Publishing Co., pp. 199 – 226, 2002.

[12] G. Kim, V. Govindaran, S. Srihari, Architecture for Handwritten Text Recognition Systems, in: S. –W. Lee (eds.), Advances in Handwritten Recognition, World Scientific Publishing. Co., pp. 163 – 172, 1999

[13] Héla Zouari, Laurent Heutte, Yves Lecourtier, Using diversity measure in building classifier ensembles for combination method analisis, Laboratoire Perception, System Information (PSI), Université de Rouen, Place Emile Blondel, 76821 Mont-Saint-Aignan Cedex, France.

[14] Internal technical Report, Automated Control and Guidance System – ACOS, University of Kent, Department of Electronics. Canterbury, Kent, CT2 7NT, U.K. July 2005.

[15] J. –C. Simon, Off-line cursive Word Recognition, Proc. Of the IEEE 80(7) , pp. 1150 – 1161, 1992

[16] Kyoung Min Kim et al., recognition of handwritten numerals using a combined classifier with hybrid features, A. fred et al. (Eds.): SSPR&SPR 2004, LNCS 3138, Springer – Verlag Heidelberg pp. 992 – 1000, 2004.

[17] L. Lam, Classifier combinations: Implementations and theoretical issues in MCS '00: proceedings of the first international workshop on multiple classifier systems, Springer-Verlag, London, UK, pp. 77 – 86 2002.

[18] L. Xu, A. Krzyzak, C. Y. Suen, Associative Switch for Combining Multiple Classifiers and their Application in Handwritten Character Recognition, IEEE Trans. On System, Man , and Cybernetics SMC – 22(3), pp. 418 – 435, 1992.

[19] M. Morita, L. S. Sabourin, F. Bortolozzi, and C. Y. Suen, Centre for Pattern Recognition and Machine Intelligence, Pontificia Universidade Catolica do Parana, IEEE, 2002.

[20] P. Lorrentz, W. G. J. Howells, K. D. McDonald-Maier, Enhanced Probabilistic Convergent Network, in: Proceedings of the 6th international conference on recent advances in Soft Computing (RASC 2006), K. Sirlantzis (Ed.), pp. 267 – 272, 2006.

[21] R. P. W. Duin, The combining classifier: to train or not to train?, ICPR2002, Quebec City, 11 – 15 of August 2002.

[22] R. Ranawana, V. Palade, Multi-classifier systems – review and a roadmap for developers, University of Oxford Computing Laboratory, April 24 2006.

[23] R. Ranawana, V. Palade, A neural network based multi-classifier system for gene identification in DNA sequences, Neural Comput. Appl. 14(2), pp. 122 – 131, 2005.

[24] S. Impedovo, P. Wang, H. Bunke (Eds.): Automatic Bankcheque Processing, World Scientific Publ. Co., Singapore, 1997.

[25] Simon Günter, Horst Bunke, Feature Selection Algorithm for the Generation of Multiple Classifier Systems and their Application to Handwritten Word Recognition, Department of Computer Science, University of Bern, Bern, Switzerland, March 4 2004.

[26] T. G. Dietterich, Machine-learning research: Four current directions, The AI magazine 18(4), pp.97 – 136, 1998.

[27] T. K. Ho: The random subspace method for constructing decision forests, IEEE Trans. On Pattern Analysis and Machine Intelligence 20(8), pp. 832 – 844, 1998.

[28] W. G. J. Howells, M. C. Fairhurst & Faud Rahman, An exploration of a new paradigm for weightless RAM-based neural networks, Electronic, Connection Science, Voln. 12, No. 1 pp. 65-90., 2000.

[29] W. G. J. Howells, M.C. Fairhurst, D.L. Bisset, PCN: The Probabilistic Convergent Network, Electronics Engineering Laboratories, University of Kent, Canterbury, Kent, CT2 7NT, U.K. November 1995.

[30] W. G. J. Howells, S. Kola, T. Statheros, K. McDonald-Maier: An Intelligent Fast-Learning Multi-Classifier System based on Weightless Neural Architectures, in: Proc. Of the Int. conf. On Recent Advances in Soft Computing (RASC 2006), K. Sirlantzis (Ed.), pp. 72 – 77, 2006.

[31] Y. Freund, R. Schapire, A descision-theoretical generalisation of on-line learning and application to boosting, Journal of computer and systems sciences 55(1), (1997), 119 – 139.