Generalized CMAC Adaptive Ensembles for Concept-Drifting Data Streams

Francisco J. González-Serrano
Department of Signal Theory and Communications,
Universidad Carlos III de Madrid,
Avda. Universidad, 30, 28911, Leganés, Spain.
Email: fran@tsc.uc3m.es

Aníbal R. Figueiras-Vidal
Department of Signal Theory and Communications,
Universidad Carlos III de Madrid,
Avda. Universidad, 30, 28911, Leganés, Spain.
Email: arfv@tsc.uc3m.es

***Abstract*—In this paper we propose to use an adaptive ensemble learning framework with different levels of diversity to handle streams of data in non-stationary scenarios in which concept drifts are present. Our adaptive system consists of two ensembles, each one with a different level of diversity (from high to low), and, therefore, with different and complementary capabilities, that are adaptively combined to obtain an overall system of improved performance. In our approach, the ensemble members are generalized CMACs, a linear-in-the-parameters network. The ensemble of CMACs provides a reasonable trade-off between expressive power, simplicity, and fast learning speed.**

At the end of the paper, we provide a performance analysis of the proposed learning framework on benchmark datasets with concept drifts of different levels of severity and speed.

I. Introduction

In many signal and data processing applications, the distribution of data may change over time resulting a concept drift [1]. The term *real concept drift* describes changes in the conditional distribution of the output given the input, while the distribution of the input may remain stable. A typical example of the real concept drift is a personalized information recommendation system: usually, the conditional distribution of the interesting (and not interesting) documents for that user may change, while the distribution of the incoming stream of multimedia documents remains constant.

Apart from filtering for recommendation systems, concept drift is a central problem in many dynamically changing and non-stationary environments, including medicine [2], industry [3], education [4], and business [5]. In these dynamic scenarios the challenge is to process, in (near) real-time, large high-speed data streams in order to adapt the learning model by combining what has been learned in the old concept with the fresh information corresponding to the new concept.

One of the most promising and effective approaches to address these challenges is online ensemble learning. In ensemble learning, a set of models are combined, usually according to their expertise level regarding the current concept. Ensemble online learning has been successfully used to improve the accuracy of single learners in different applications such as classification, regression, time series prediction, and filtering [6].

Our approach is based on the use of diversity to improve the ensemble learning when concept drifts are present. Diversity reflects the degree of agreement between base learners in the ensemble. When pairs of base learners tend to agree, the ensemble is considered less diverse. When an ensemble is highly diverse, a *gradual* change of concept does not necessary imply that the base learners themselves are no longer useful. In fact, as the base learners are “different”, it is possible that some of them will already be useful for solving the new concept. On the contrary, low diversity ensembles converge faster and are usually the most accurate when the drift causes very big changes and occurs suddenly (*abrupt* concept drift).

Since concept drifts can have different speeds and severities, it seems reasonable to dinamically combine high and low diversity ensembles to improve the adaption to the new concept. Following this approach, we propose to use an adaptive convex combination of two high and low diverse ensembles. The obtained results show that the mixing parameter automatically selects the best combination in different concept drift scenarios.

II. Concept Drift

From a formal point of view, a concept drift can be defined as $p^{t_1}(\boldsymbol{x}, y) \neq p^{t_0}(\boldsymbol{x}, y)$ where p^{t_i} is the joint probability distribution at time t_i between the set of input variables $\boldsymbol{x}$ and the target variable y. A change in the data distribution may occur because there may be a change in:

- the prior probabilities of classes $p(y)$,
- the class conditional probabilities $p(\boldsymbol{x}|y)$, and
- as a result, the posterior probabilities of classes $p(y|\boldsymbol{x})$.

In particular, a *real concept drift* refers to changes in $p(y|\boldsymbol{x})$, which can happen either with or without a modification in the input data distribution $p(\boldsymbol{x})$.

Concept drifts can be described in terms of their severities and speeds [7]. Severity is the amount of changes caused by a new concept. Typically is expressed as the percentage of the input space which has changed once the drift is completed. Speed is the inverse of drifting time, and it can be defined as the length of the time interval that the new concept needs to completely replace the old one. According to the speed, drifts can be categorized as either abrupt (high speed), when the complete change occurs in just one time step, or gradual (low speed), otherwise.

ISBN 978-0-9928626-7-1 © EURASIP 2017

III. A Solution for Concept Drift

In this paper, we propose a new approach consisting of an adaptive convex combination of the low and high diversity ensembles. The ensembles are combined in such a manner that the advantages of both of them are kept: the rapid convergence from the fast low diversity ensemble, and the reduced steady-state error from the highly diverse one. In analogy of a well-known neurological fact: human brains combine fast and coarse reactions against abrupt changes in the environment, with an early processing at the amygdala, and more elaborated but slower responses taken in the neocortex at a conscious level [8].

A. Ensembles for concept drift: local and diverse

The relationship among diversity, accuracy on the training set, and generalization is complex, especially in online changing environments. In particular, the accuracy of ensembles depends on the base learner, their diversity [9] and the level of severity and speed of drift present in the dataset.

Regarding the base learners, an important problem is that the data distribution may change only over a constrained region of the input space (*local* concept drift). One example in the real world is spam filtering, where only some particular types of spam may change with time, while the others could remain the same. In these cases, the accuracy of global models may fall, even if they still could be good experts in the stable parts of the data [2].

From earlier research work on concept drift [7], it has been found that high diversity ensembles provide the best generalization accuracy for drifts at low speed. For high speed and high severity, it is a good strategy to use low diversity ensembles. If the dataset contains drifts with low severity (and high speed), high diversity ensembles would have problems to converge to the new concept. However, low diversity ensembles are able to converge rapidly to the new one.

In our approach, the degree of diversity in an ensemble is controlled with Online Bagging [10]. In Online Bagging, training examples are sampled with replacement, so a base learner can be updated k-times using a newly arrived instance. The value of k is selected according to a Poisson distribution, where $k \sim$ Poisson (λ). Different levels of diversity in an ensemble are explicitly introduced by varying the λ: higher/lower values of λ are associated with lower/higher diversity in an ensemble of experts.

B. Convex combination

The basic idea behind convex combination is that two (or more) adaptive models (or ensembles), with complementary capabilities, adaptively combine their outputs by means of a mixing parameter, to obtain an overall model of improved performance:

$$y(t) = \beta(t)y_1(t) + [1 - \beta(t)]y_2(t), \tag{1}$$

where $y_i(t)$, i = 1, 2 are the outputs of the component models, $y(t)$ is the overall output, $\beta(t)$ is a mixing parameter in the range (0, 1), and t is the time index. If $\beta(t)$ is appropriately updated, it can be shown that the resulting model performs as well as or better than the best individual component under certain conditions [11]. The adaptation rule for $\beta(t)$ is obtained as follows:

$$a(t+1) = a(t) + \frac{\mu_a}{p(t)} e(t)[e_2(t) - e_1(t)]\lambda(t)[1 - \lambda(t)], \tag{2}$$

where $e(t) = d(t) - y(t)$ and $e_i(t) = d(t) - y_i(t)$, i = 1,2, are the errors of the overall model and the components, $d(t)$ being the desired signal, μ_a is a step-size parameter, $p(t) = \gamma p(t-1) + (1-\gamma)[e_2(t) - e_1(t)]^2$ is a power normalizing sequence that improves convergence [12], and $a(t)$ is a parameter that defines $\beta(t)$ via a sigmoidal function as:

$$\beta(t) = \mathrm{sgm}[a(t)] = \frac{1}{1 + e^{-a(t)}} \tag{3}$$

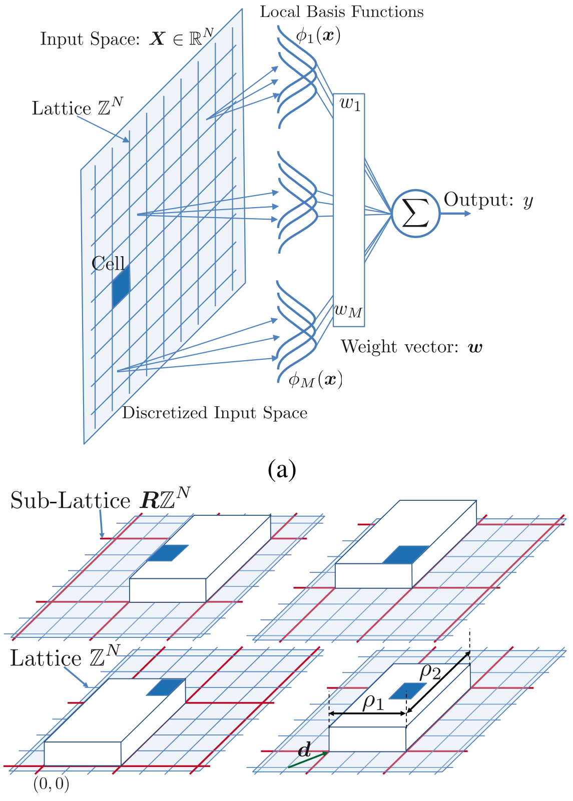
C. Base learner: Generalized CMAC

As it has been mentioned before, a proper selection of the base learner may have a significant effect on the performance of the combined ensembles. We propose to use the Cerebellar Model Articulation Controller (CMAC) [13] as the base learner for the following reasons. The CMAC is a very simple linear-in-the-parameters network (see Figure 1.a), yet with a high expressive capability, being able to approximate, with a high degree of accuracy, a large number of functions just by choosing a single parameter (the generalization factor) [14]. Additionally, the CMAC uses a set of overlapping local basis functions, which can provide good performance in some practical applications where local concept drifts are present (see Section III-A). Because of its fast online training and simplicity, it is very useful in many real-time applications such as control (its original purpose) [15], predistortion [16], classification [17], and time series forecasting [18].

The CMAC network performs a local approximation of functions by means of a weighted combination of overlapping local basis functions. The position and size of CMAC's basis functions are predetermined at the initialization. Unlike the conventional CMAC, the architecture of the Generalized CMAC (GCMAC) [19] depends on a integer-valued generalization vector $\boldsymbol{\rho} = [\rho_1, \dots, \rho_N]^T \in \mathbb{N}^N$ that defines local basis functions with different widths, and, therefore, with different degrees of generalization, in each input.

The domains of the basis functions are defined by using $\rho_{\max}$ linear manifolds, or cosets, of $\mathbb{Z}^N$ of the form $\boldsymbol{R}\mathbb{Z}^N + k\boldsymbol{d}$, $k = 0, \dots, \rho_{\max} - 1$, where $\boldsymbol{R}\mathbb{Z}^N$ is the sublattice defined by $\left\{\boldsymbol{x} \in \mathbb{Z}^N / \boldsymbol{x} = [q_1\rho_1, \dots, q_N\rho_N]^T\right\}$, and $\boldsymbol{d} = [d_1, \dots, d_N]^T$, the displacement vector. The elements of the displacement vector must be selected ensuring that $\boldsymbol{d}$ and $\boldsymbol{\rho}$ are coprimes, in order to guarantee that the set of basis functions activated by two adjacent input points differs at least in one element. The linear manifolds divide the input space into overlapped hyper-rectangular regions of size given by $\boldsymbol{\rho}$, which constitute the domains of the basis functions (see Figure 1.b).

Regarding the shape of the basis functions, the original Albus' CMAC uses constant functions, although other functions such as B-splines [20] and Gaussian functions [21] can be also defined.



(a)

(b)

Fig. 1: The GCMAC network. (a) Three-layer architecture. (b) Constant LBFs.

The output of the M basis functions can be arranged into an activation vector, $\boldsymbol{\phi}(\boldsymbol{x})$, that contains only $\rho_{\max}$ nonzero values. The GCMAC output is computed as $y_{\text{GCMAC}} = \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{w}$, where $\boldsymbol{w}$ is the weight vector.

The weight vector $\boldsymbol{w}$ can be trained using a simple first-order learning rule in which the error is divided in equal parts among the $\rho_{\max}$ functions participating in the output (Albus' rule). A faster rule is to use a variable learning step that depends on the mean power of the association vector:

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \frac{\mu_G}{D}(y - y_{\text{GCMAC}})\boldsymbol{C}\boldsymbol{\phi}(\boldsymbol{x}), \tag{4}$$

where $D = \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{C}\boldsymbol{\phi}(\boldsymbol{x})$, and $\boldsymbol{C}$ is a diagonal matrix with entries equal to the mean power of vector $\boldsymbol{\phi}(\boldsymbol{x})$.

D. Our proposal

Our proposal is depicted in Figure 2. We have included time-delayed connections (a Shift Register) for sequential information processing. It is important to note that our scheme does not include an explicit (and possibly complex) concept drift detection method. Instead, the handling of the non-stationary behavior is done by the (more simple) convex combination of ensembles.

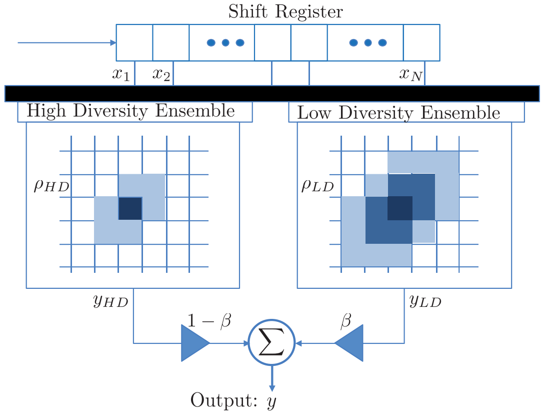


Fig. 2: Convex combination of low/high diversity ensembles using GCMAC as base learner.

IV. Results

In order to analyze the effect of diversity in the presence of concept drift, we have used the artificial dataset "Circle", defined as: $(x_1 - a)^2 + (x_2 - b)^2 \gtrless r^2$ ($a = b = 0.5$; $\boldsymbol{x} \in [0, 1]^2$) [7]. To simulate the drift, the circle radius r changes from $r = 0.2$ to 0.3 (*low severity*), or to $r = 0.5$ (*high severity*). In the old concept, $L = 2000$ samples have been generated: those falling inside the circle are labelled as $+1$ (outside, -1; balanced samples); after the drift, labels are swapped, and samples inside the circle are labelled as -1. In *high speed* drifts, the concept changes abruptly in $t = L + 1$; in *low speed* drifts, the gradual change from the old to the new concept lasts 1000 samples (see Figure 3). Finally, to simulate noise, 5 % of samples have been wrongly labelled.

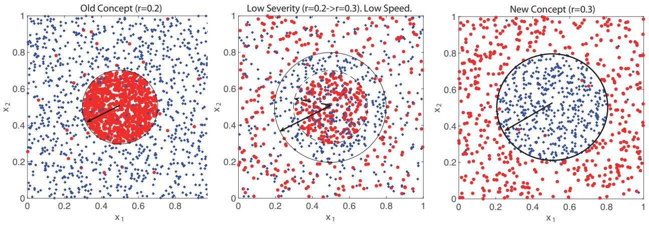


Fig. 3: Concept drift using the "Circle" dataset ($a = b = 0.5$). The drift from the old concept ($r = 0.2$) to the new concept ($r = 0.3$) takes place at Low Speed.

The performance is analyzed in terms of the *prequential accuracy*, $acc(t)$, defined as the average accuracy computed from each example presented for training, prior to the example being learned. In order to analyze the behavior of the ensembles before and after the beginning of a drift, the accuracy is reset when the drift starts ($t = L + 1$). The depicted values of prequential accuracy have been obtained after a total of 30 independent runs.

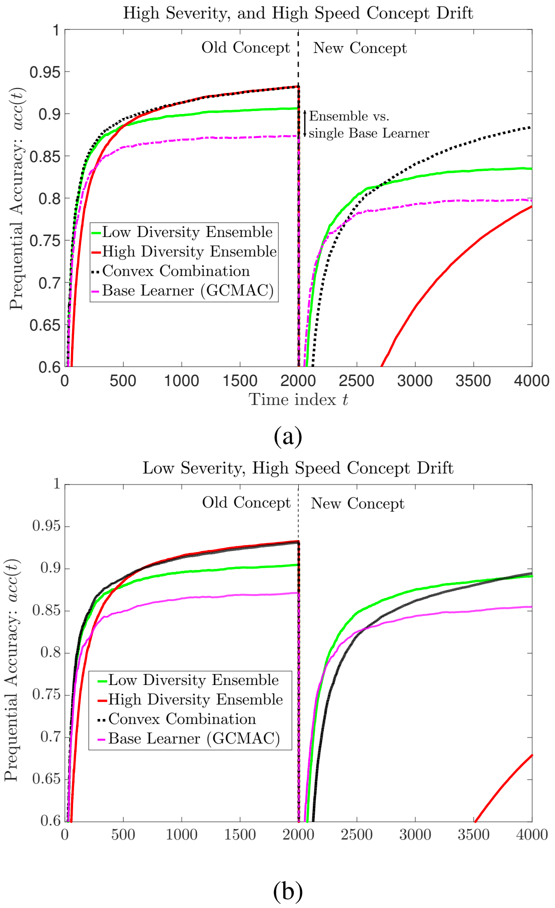
The Poisson(λ) distribution for a low diversity ensemble uses $\lambda_{LD} = 1$, and the value for the high diversity ensemble is $\lambda_{HD} = 0.05$.

ISBN 978-0-9928626-7-1 © EURASIP 2017

In the experiments we have used 25 GCMAC networks as the base learners for each ensemble. The input space was discretized with an 8-bit quantizer. A generalization factor $\rho_{LD} = 255$ was used in the low diversity ensemble, and $\rho_{HD} = 32$ in the high diversity one (see Figure 2). A step size $\mu_G = 1$ was used in the learning rule (4).

Finally, in the convex combination we have used a step size $\mu_a = 1$ and a forgetting factor $\gamma = 0.9$ (see Equation (2)).

The evolution of the prequential accuracy of the low and high diversity ensembles, the convex combination, and the single base learner for a High Speed Concept Drift is depicted in Figure 4.



(a)

(b)

Fig. 4: Evolution of the prequential accuracy of the low and high diversity ensembles, the convex combination, and the single base learner for a High Speed concept drift. (a) High Severity. (b) Low Severity

From Figure 4.a (High Severity), some conclusions may be drawn. The first one is the capacity of an ensemble to improve upon the accuracy of the base learner. Second, the low diversity ensemble has a faster reaction time than the high diversity ensemble. And third, our convex combination clearly outperforms the individual ensembles after the drift.

A similar behavior is observed for Low Severity abrupt drifts (see Figure 4.b). However, the performance of the high diversity ensemble is now very poor (close to random guessing). A possible explanation is that the high diversity ensemble retains the learned old concept for a long time because it is unable to detect a slight concept drift.

These behaviors are implicitly considered in the convex combination of ensembles. Analyzing the evolution of the mixing parameter $\beta(t)$ (see Figure 5), it is observed that the Low Diversity ensemble is preferred for a long time ($\beta(t) > \frac{1}{2}$) when the severity of the concept drift is low.

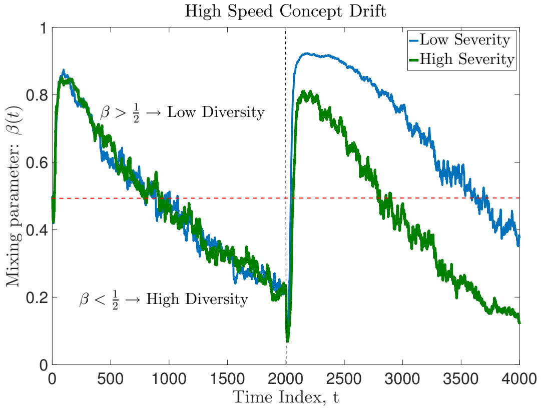


Fig. 5: Evolution of $\beta(t)$ for a High Speed concept drift.

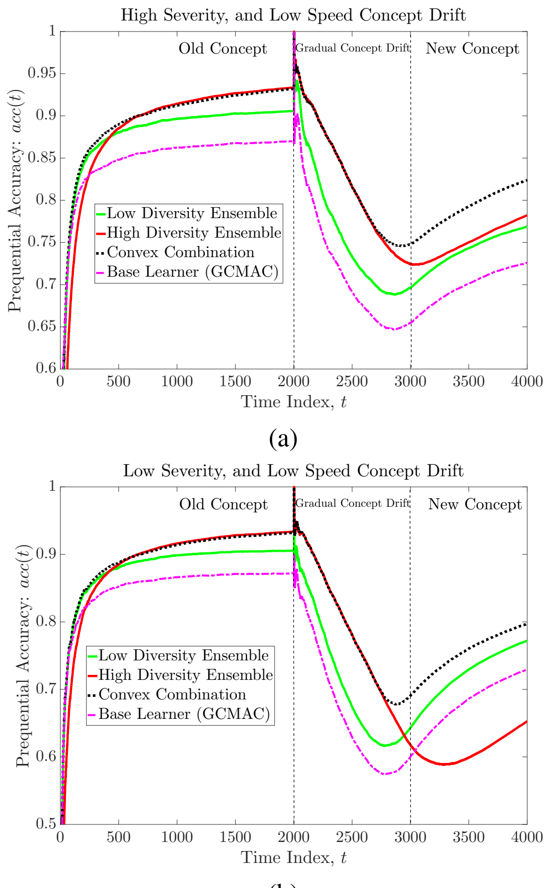
The prequential accuracy for Low Speed drifts is depicted in Figure 6, and the evolution of the mixing parameter $\beta(t)$ is depicted in Figure 7.

The analysis of this results leads to the following conclusions. Independently of the severity of the gradual drift, the High Diversity ensemble is the best option along its duration ($\beta(t) \ll \frac{1}{2}$, for $2000 \lesssim t \lesssim 3000$). An interpretation of this could be that the more complex (and richer) architecture of the High Diversity ensemble is able to use the information learned in the old concept at the same time that learns the new concept. Once the gradual drift is completed, the choice of one ensemble or the other depends on the severity of the drift: after a high severity drift, the convex combination prefers the high diversity ensemble, and the low diversity ensemble, otherwise.

V. Conclusions

In this paper we have proposed an adaptive ensemble learning framework with different levels of diversity to handle streams of data in dynamic non-stationary scenarios where concept drifts are present. Our adaptive system consists of two ensembles, each one with different level of diversity (from high to low), and, therefore, with different and complementary capabilities, that are adaptatively combined to obtain an overall system of improved performance. We have used the generalized CMAC as the base learner due to its fast learning speed and good generalization performance.

Although more research is needed (to analyze the behavior of new adaptive convex combination strategies on real



(a)

(b)

Fig. 6: Evolution of the prequential accuracy for a Low Speed concept drift. (a) High Severity. (b) Low Severity

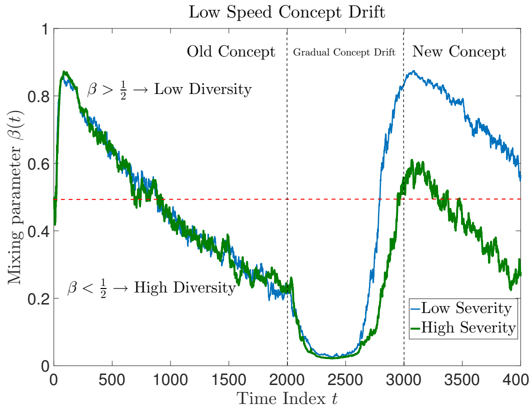


Fig. 7: Evolution of $\beta(t)$ for a Low Speed concept drift.

datasets), the obtained results show that the convex combination of high and low diversity ensembles achieves good performance, in terms of accuracy and tracking capabilities, on datasets with concept drifts.

Acknowledgment

This work is partially funded by grant CASI-CAM-CM (S2013/ICE-2845), DGUI-Comunidad de Madrid, and grants DAMA (TIN2015-70308-REDT), MINECO, and Macro-ADOBE (TEC 2015-67719-P), MINECO-FEDER-EU.

References

[1] J. C. Schlimmer and R. H. Granger, "Beyond incremental processing: Tracking concept drift." in *AAAI*, 1986, pp. 502–507.

[2] A. Tsymbal, M. Pechenizkiy, P. Cunningham, and S. Puuronen, "Dynamic integration of classifiers for handling concept drift," *Information Fusion*, vol. 9, no. 1, pp. 56–68, 2008.

[3] I. Žliobaitė, M. Pechenizkiy, and J. Gama, "An overview of concept drift applications," in *Big Data Analysis: New Algorithms for a New Society*. Springer, 2016, pp. 91–114.

[4] G. Castillo, J. Gama, and A. M. Breda, "An adaptive predictive model for student modeling," *Advances in Web-based education: Personalized learning environments*, vol. 7092, 2005.

[5] M. Scholz and R. Klinkenberg, "Boosting classifiers for drifting concepts," *Intelligent Data Analysis*, vol. 11, no. 1, pp. 3–28, 2007.

[6] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1517–1531, 2011.

[7] L. L. Minku and X. Yao, "DDD: A new ensemble approach for dealing with concept drift," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 4, pp. 619–633, 2012.

[8] J. Arenas-García, M. Martínez-Ramón, Á. Navia-Vázquez, and A. R. Figueiras-Vidal, "Plant identification via adaptive combination of transversal filters," *Signal Processing*, vol. 86, no. 9, pp. 2430–2438, 2006.

[9] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.

[10] N. C. Oza, "Online bagging and boosting," in *Systems, Man and Cybernetics, 2005 IEEE international conference on*, vol. 3. IEEE, 2005, pp. 2340–2345.

[11] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1078–1090, 2006.

[12] L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-García, "A normalized adaptation scheme for the convex combination of two adaptive filters," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2008, pp. 3301–3304.

[13] J. Albus, "A new approach to manipulator control: the Cerebellar Model Articulation Controller," *Journal on Dynamic Systems, Measurements and Control*, vol. 63, no. 3, pp. 220–227, 1975.

[14] M. Brown, C. J. Harris, and P. C. Parks, "The interpolation capabilities of the binary CMAC," *Neural Networks, Pergamon Press Ltd.*, vol. 6, no. 3, pp. 429–440, 1993.

[15] C. M. Lin and T. Y. Chen, "Self-organizing CMAC control for a class of MIMO uncertain nonlinear systems," *IEEE Transactions on Neural Networks*, vol. 20, no. 9, pp. 1377–1384, 2009.

[16] F.-J. González-Serrano, J. J. Murillo-Fuentes, and A. Artés-Rodríguez, "GCMAC-based predistortion for digital modulations," *IEEE Transactions on Communications*, vol. 49, no. 9, pp. 1679–1689, 2001.

[17] J.-Y. Wu, "MIMO CMAC neural network classifier for solving classification problems," *Applied Soft Computing*, vol. 11, no. 2, pp. 2326–2333, 2011.

[18] C.-J. Lu and J.-Y. Wu, "An efficient CMAC neural network for stock index forecasting," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15 194–15 201, 2011.

[19] F. J. González-Serrano, A. Artés-Rodríguez, and A. R. Figueiras-Vidal, "Generalizing CMAC architecture and training," *IEEE Transactions on Neural Networks*, vol. 9, no. 6, 1998.

[20] S. H. Lane, D. A. Handelman, and J. J. Gelfand, "Theory and development of higher-order CMAC neural network," *IEEE Control Systems*, vol. 12, no. 2, pp. 23–30, 1992.

[21] C.-T. Chiang and C.-S. Lin, "CMAC with general basis functions," *Neural Networks*, vol. 9, no. 7, pp. 1199–1211, 1996.