On the Creation of Diverse Ensembles for Nonstationary Environments using Bio-inspired Heuristics

Jesus L. Lobo[1], Javier Del Ser[1,2,3], Esther Villar-Rodriguez[1], Miren Nekane Bilbao[2], and Sancho Salcedo-Sanz[4]

[1] TECNALIA, E-48160 Derio, Spain,
{jesus.lopez,javier.delser,esther.villar}@tecnalia.com
[2] University of the Basque Country UPV/EHU, 48013 Bilbao, Spain,
{javier.delser,nekane.bilbao}@ehu.eus
[3] Basque Center for Applied Mathematics (BCAM), 48009 Bilbao, Spain
[4] Universidad de Alcalá, E-28871 Alcalá de Henares, Spain,
sancho.salcedo@uah.es

Abstract. Recently the relevance of adaptive models for dynamic data environments has turned into a hot topic due to the vast number of scenarios generating nonstationary data streams. When a change (concept drift) in data distribution occurs, the ensembles of models trained over these data sources are obsolete and do not adapt suitably to the new distribution of the data. Although most of the research on the field is focused on the detection of this drift to re-train the ensemble, it is widely known the importance of the diversity in the ensemble shortly after the drift in order to reduce the initial drop in accuracy. In a Big Data scenario in which data can be huge (and also the number of past models), achieving the most diverse ensemble implies the calculus of all possible combinations of models, which is not an easy task to carry out quickly in the long term. This challenge can be formulated as an optimization problem, for which bio-inspired algorithms can play one of the key roles in these adaptive algorithms. Precisely this is the goal of this manuscript: to validate the relevance of the diversity right after drifts, and to unveil how to achieve a highly diverse ensemble by using a self-learning optimization technique.

Keywords: Concept Drift; Diversity; Bioinspired optimization

1 Introduction and Related Work

The increasing number of applications favoring the generation of data streams – such as mobile phones, sensor networks and in general all scenarios under the so-called *Internet of Things* paradigm [1] – has led the research community to the necessity for new approaches capable of dealing with fast incoming information flows. In these practical situations it is often assumed that the process behind the generation of such data streams is stationary, i.e. the statistical properties

of the underlying phenomena that produce the information to be processed do not vary along time. Unfortunately, in many real scenarios this assumption does not hold since the data generation process becomes affected by a nonstationary event (such as eventual changes in the users' habits, seasonality, periodicity, sensor errors, etc.). Under these circumstances the statistical distribution of the data may change (drift), which ultimately causes that models trained over these data sources are obsolete and do not adapt suitably to the new distribution of the data. Therefore, in the context of data mining in such nonstationary environments the construction of learning models requires adaptive approaches to ease the adjustment of such model to drifts, either from an active (i.e. drift detection, which triggers a subsequent model adaptation) or a passive perspective (the adaptation of the model whenever new data arrive).

Ensembles are one of the most useful approaches to deal with concept drift, and have been successfully used to improve the accuracy of single classifiers in incremental learning. Diversity among the constituent learners in ensemble models has been empirically proven to be crucial when dealing with concept drift [2]. Specifically this study evinces that the diversity plays an important role *before* and *after* a concept drift, importance that is also subject to the severity of the drift: before the drift, ensembles with less diversity obtain better test errors, while shortly after the drift more diverse ensembles use to score lower test error rates. Their difference in terms of test error performance when compared to lower diversity ensembles is usually more significant when the severity is higher. Therefore, it is a good strategy to maintain highly diverse ensembles and utilize them shortly after the drift (independently from the type of drift) to obtain good performance scores. The so-called Diversity for Dealing with Drifts (DDD) approach published in [3] leverages this empirically validated conclusion, and is one of the most recognized methods to manage diverse ensemble in the presence of concept drift from an active perspective.

Due to the above noted importance of achieving a good balance between adaptability (diversity) and performance along time, there is a latent need for novel mechanisms to optimally balance the diversity in ensemble learning. This work falls within this research trend and formulates the diversity balance as an optimization problem. We explore the benefits of a bio-inspired solver for the construction of ensembles with different levels of diversity, in particular the Harmony Search (HS) algorithm [4]. HS has demonstrated to be competitive respect to other evolutionary heuristics for optimization paradigms in diverse fields such as energy [5,6], bio-informatics [7], telecommunications [8,9], data mining and concept drift [12], and logistics [13], among many others [10,11]. However, to the knowledge of the authors no previous contribution has gravitated on the diversity-accuracy trade-off in ensemble learning over nonstationary data.

The idea behind this research work is to use the HS algorithm to build ensembles of models with maximum and minimum diversity, and then utilize them shortly after the drift to show that ensembles with high diversity yield better classification performance those with low diversity. In this regard it has been widely acknowledged in the literature (see e.g. [14,15,16] and references therein)

that the Area Under the ROC Curve (AUC) is the most strongly recommended score due to the fact that the naive accuracy metric is not a reliable indicator in severely imbalanced data sets. This work will embrace this recommendation in what follows, specially in Section 4 for comparing results among different ensembles.

The rest of the paper is organized as follows: Section 2 introduces the analyzed scenario. Section 3 delves into the proposed approach based on HS, whereas Section 4 presents and discusses the simulation results obtained over the SEA data set [20]. Finally, Section 5 ends the paper and sketches future research lines.

2 Analyzed Scenario

In batch learning [15] the level of diversity among base learners in the ensemble is a relevant topic that has grasped notable attention in the literature. The success of ensemble learning algorithms is based, to a certain point, on the accuracy and diversity among the base learners [17]; some studies have revealed that it exists a positive correlation between accuracy of the ensemble and diversity among its members [18]. In [2] it was concluded that it is a good strategy to maintain highly diverse ensembles to obtain good responses shortly after the drift, independently of the type of the drift. However, at this point it is necessary to choose a metric to measure the diversity of an ensemble so as to build ensembles with different levels of diversity depending on the instant at which it is applied (before or shortly after the drift).

Following the recommendations of [18], in which a thorough analysis of 10 measures was discussed, the Yule's Q statistic [19] is selected for the purpose of minimizing the error of ensembles. The advantages of this measure are its simplicity and ease of interpretation. Considering two classifiers C_i and C_j, the Yule's Q statistic metric can be calculated as

$$Q_{i,j} \doteq \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}, \tag{1}$$

where N^{ab} is the number of training samples for which the classification given by C_i is a and the classification given by C_j is b. We further assume that 1 represents a correct classification and 0 is a misclassification. Q varies between -1 and 1. Classifiers that tend to recognize the same objects correctly will have positive values of Q, and those which commit errors on different objects will render Q negative. For an ensemble E of M classifiers, the $\bar{Q}$ statistic averaged over all pairs of classifiers is given by

$$Q_{averaged} \doteq \frac{2}{M(M-1)} \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} Q_{i,j}. \tag{2}$$

where, as mentioned above, higher/lower $\bar{Q}$ values are associated with lower/higher diversity, establishing an inversely proportional relation.

Bearing these definitions in mind a batch learning technique based on an ensemble composed of several base learners has to deal with applications that provide fast incoming information flows in the form of data batches. In this scenario one different model can be trained with each newly incoming batch, hence the total number of combinations that may yield possible diverse ensembles at a concrete time step can be too high in the long run for its exhaustive evaluation. Due to the time constraints or computation costs in certain cases of these applications, the task of finding diverse ensembles may not be affordable in practice. For this reason this challenge can be dealt with by formulating the choice of diverse base learners as an optimization problem, for which a bio-inspired technique can find an optimal diverse ensemble at each moment on time.

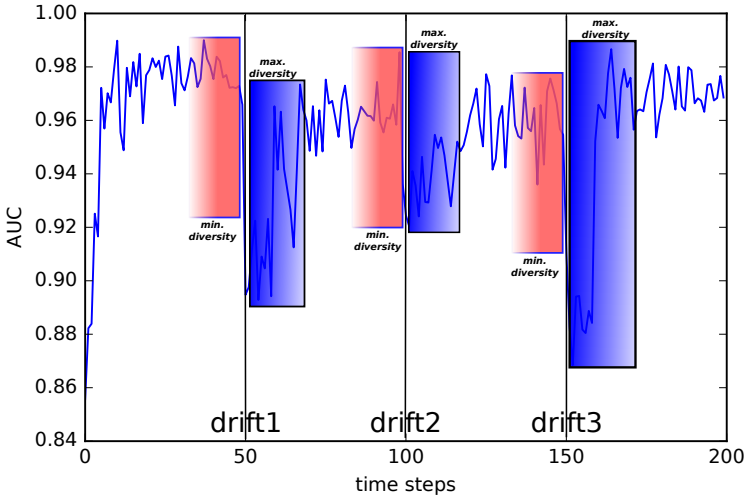

Fig. 1. Diversity importance before and after the drift for the SEA data set.

In the case of the online learning approach proposed in [3], the learning process of the ensemble is carried out for a fixed number of times (defined by the rate parameter λ characterizing the $Poisson(\lambda)$ distribution) with the same current training data. As it is not possible to store past data, so the ensemble can not learn from past information. In this way, higher/lower λ values are associated with higher/lower $\bar{Q}$ values (lower/higher diversity). In the case of a batch learning approach, a model can be trained with incoming training examples, and be part of the ensemble if it is considered, being it possible to have an ensemble formed by members trained in the past. This work follows a batch learning approach, and uses an HS solver to maximize and minimize the diversity of the ensemble. Figure 1 represents a batch learning process during 200 time steps with the AUC score, and it shows how the importance of the diversity is at each moment before and after the drift for the SEA dataset [20].

In order to test the feasibility of HS to achieve different levels of diversity for the ensemble, this work has been evaluated when applied over one of the most widely used synthetic data sets for assessing new concept drift developments:

the SEA data set. Following the original data set generation procedure posed in this work, a total of 10000 3-dimensional samples have been generated at random within the range $\mathbb{R}[0, 10)$. Only the first two dimensions (features) are set informative for the class to be predicted, whereas the remaining dimension is irrelevant and acts as a noisy component for the target label. Points have been split in 200 batches of length 250 samples, which have been further divided into 4 main groups or blocks characterized by different concepts: a data sample belongs to class 1 if $x_1 + x_2 \leq \Theta$, where x_1 and x_2 represent the first two features of the sample and Θ is a threshold value that sets the frontier between the two classes. A recurrent series of values (i.e. $\Theta = \{4, 7, 4, 7\}$) has been used to generate the four concept blocks. An additional class noise has been also inserted within each block by randomly changing the class of 5% of the total instances.

3 Proposed HS-based Approach

HS works by imitating the activity of musicians while improvising new music pieces. The choice of which note to play next is something which takes years to learn to do effectively. Each musician in the band (ensemble) is often faced with the problem of picking the next note. To do so they can resort to their knowledge of the notes in the key they are playing in (which notes sound aesthetically pleasant in the context of the song), as well as the notes they have played previously (what notes sounded good in the recent context). The notes they played recently are most likely to sound pleasantly. Also, it is wise to select a particular note that the audience might expect and adjust the pitch ignoring the expected note to create an artistic effect and a new, potentially better, harmony. HS seeks an optimal combination of inputs, just as a musician seeks a good harmony. HS generates "harmonies" of inputs which are evaluated for quality, and iterates this process until it finds the best one possible. The aesthetics of a musical harmony are analogous to the fitness of a particular solution, so following this simile HS attempts to achieve a good combination of inputs, just like musicians optimize their note selection using their own heuristics. Each input to the problem is considered as a different instrument in an ensemble, each potential note corresponds to each potential value of the inputs that the function might adopt. The musical harmony of notes is modeled as a programmatic harmony of values. Each iteration a new harmony is generated its quality is calculated: if it makes the cut it is included in the musician's memory. This way, iteration by iteration, old poor quality harmonies are discarded and replaced by better ones. The average quality of the set of harmonies in this memory as a whole gradually increases as these new harmonies replace poor ones.

This being said, notes in the HS solver particularized to the problem tackled in this paper represent the members of the ensemble. At each time step t_i, HS optimizes the diversity of the ensemble formed by 10 members, combining all past models trained from t_0 to t_{i-1}. Special attention deserves the fact that the more past time steps are handled at the time where the ensemble is to be built, the more necessary an efficient optimization technique is, because there are more

candidates (models) to form the ensemble. In the last time step there are 199 different models, thus considering that the order of the selected models does not matter and that a model can not be selected more than once, there are 2.13×10^{16} possible combinations (being $n \doteq 199$ and $m \doteq 10$) to form an ensemble of 10 base learners at this time. Taking into account that a Big Data application may have millions of time steps (and models), the need for an optimization technique can be solidly argued.

Table 1. HS similarities in the proposed approach

Element	HS original definition	Proposed approach
Instrument	One of the inputs to the quality function	Each ensemble at time step t_i composed of 10 possible models from t_0 to t_{i-1}
Note	One of the possible values of an input	$Q_{averaged}$ value for each ensemble
Harmony	A combination of each instrument playing a particular note	The formation of an ensemble composed of trained models
Quality	A quantitative measure of a harmony's desirability	The Yule's Q diversity metric
Harmony Memory	The collection of good harmonies stored in memory	The collection of ensembles
HMCR	The process of generating a new harmony using random notes from the memory	The probability of choosing a model (note) of the former ensemble (harmony) for the new one
PAR	The process of moving a particular note of an instrument up or down	The probability of choosing a "similar" model (note) to the current one from the new ensemble (new harmony)

The superior performance of HS over other solutions finds its roots in their operators; the search process of HS is controlled by three different operators iteratively applied to a set of candidate solutions [4]. In a nutshell, the Harmony Memory Considerate Rate (HMCR) operator generates a new harmony using random notes from the rest of harmonies in the memory, whereas the Pitch Adjustment Rate (PAR) mutates a particular note of an instrument to a value of the vicinity of its previous value. Table 1 shows the similarities between the original definition of HS and the proposed approach.

The HS approach is applied over a total of 200 data batches, with $\mathbf{X}_{tr}(t)$ and $\mathbf{X}_{tst}(t)$ being composed by 250 samples. Every 50 batches a concept drift occurs, with 3 drift events in total. All base learners are Decision Trees and the ensemble is of size $M = 10$. This work has followed an active approach (using "perfect" drift detection) that triggers the selection of those ensembles that are more appropriate for each time slot. The study will show how to achieve an optimal level of diversity for the ensemble in each moment by the use of HS, evidencing that shortly after the drift an ensemble with high diversity obtains a better

classification performance (AUC score) than an ensemble with low diversity. HS has been used to minimize the $\bar{Q}$ metric during 10 time steps after the drift.

4 Experiments and Results

The main purpose of this work is to demonstrate the feasibility of using HS to build ensembles with maximum and minimum diversity specifically when the learning process requires it due to detected drift statistics by an external detector. In this case, Figure 1 shows three drift moments at time steps 50, 100, and 150; as already explained in Section 2, it should be a good strategy to maintain highly diverse ensembles to obtain good classification performance (AUC score) shortly after the drift. The experiments discussed in what follows aim at corroborating this recommendation first posed in [2] by means of an HS-based selection of learners for the ensemble.

In order to avoid using complex algorithm for adaptive learning (which falls out of the scope of this study), we have built ensembles of size $M = 10$ by following two different perspectives. The first refers to the ensemble formed by the best M past learners (hereafter labeled as `BEST`), i.e. those M past learners that obtain the best AUC score, being trained with their training batch data at their time but testing the current batch training data. In the second approach (corr. `LAST`) the ensemble is built with the M last learners again trained with their training batch data at that time, but testing the current batch training data. It is assumed that diversity can help mainly to reduce the initial increase in the error caused by drifts at time steps 50, 100, and 150.

Table 2. Mean AUC and mean $\bar{Q}$ scores over 15 Monte Carlos for the `BEST` approach.

		High diversity ensemble	**Low diversity ensemble**
After drift 1	AUC	0.911 ± 0.020	0.910 ± 0.025
	$\bar{Q}$	0.507 ± 0.038	0.839 ± 0.021
After drift 2	AUC	0.932 ± 0.008	0.923 ± 0.009
	$\bar{Q}$	0.495 ± 0.029	0.835 ± 0.019
After drift 3	AUC	0.894 ± 0.022	0.848 ± 0.036
	$\bar{Q}$	0.469 ± 0.036	0.837 ± 0.024

This experiment maximizes or minimizes the diversity of the ensemble after drifts 1, 2 and 3 by using HS during the period of 50-59 time steps, 100-109 time steps, and 150-159 time steps, respectively. After that, the AUC scores in these periods averaged over 15 Monte Carlo iterations are compared to confirm that it is a good strategy to have an ensemble with high diversity shortly after the drift. The HS algorithm has been configured as follows: 500 improvisations, a harmony memory size of 50 candidate solutions, a HMCR of 0.5 and a PAR of 0.1. As the role of a high diversity ensemble becomes progressively less important after

the drift, it is assumed in this work that after 10 time steps high diversity is no longer recommended to have a positive impact in the AUC score. Next, the results of the experiment are discussed.

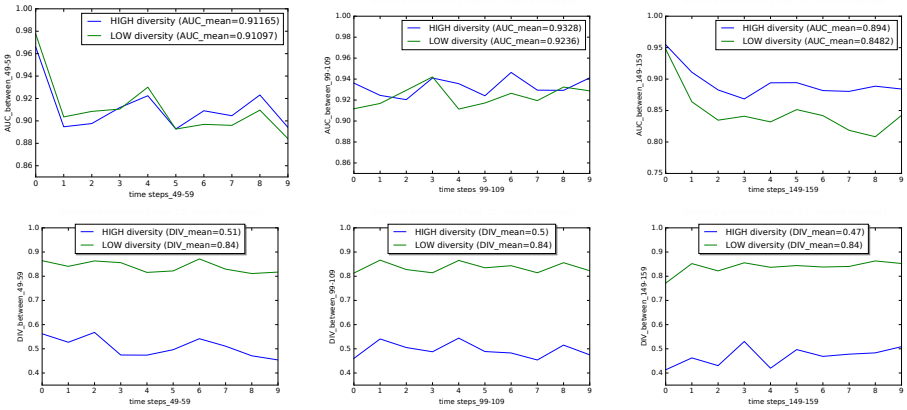

Fig. 2. Mean AUC and $\bar{Q}$ over 15 Monte Carlos after drifts 1, 2, and 3 respectively for the `BEST` approach.

As it is shown in Table 2, for the "10 best past learners" perspective, the best AUC scores are achieved when the diversity of the ensemble is maximized, in contrast with the version in which the diversity is minimized. Considering the results of the corresponding period of time, the Figure 2 shows the AUC scores of the high and low diversity ensembles, displaying a better behavior after the drifts in the case of the high diversity one. This also makes sense when checking the level of diversity at each time step.

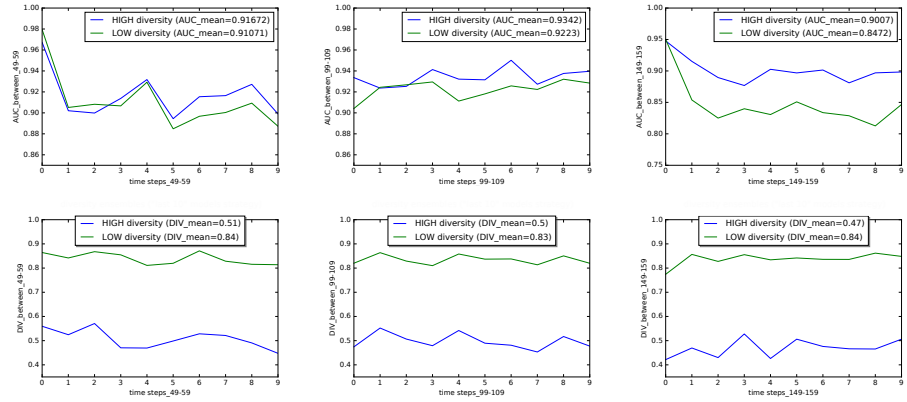

Fig. 3. Mean AUC and $\bar{Q}$ over 15 Monte Carlos after drifts 1, 2, and 3 for the `LAST` approach.

In the `LAST` case the same results are shown in Table 3, and Figure 3 shows the best performance for the high diversity ensemble.

Table 3. Mean AUC and mean $\bar{Q}$ scores over 15 Monte Carlos for the `LAST` approach.

		High diversity ensemble	**Low diversity ensemble**
After drift 1	AUC	0.916 ± 0.020	0.910 ± 0.025
	$\bar{Q}$	0.508 ± 0.037	0.838 ± 0.022
After drift 2	AUC	0.922 ± 0.008	0.934 ± 0.007
	$\bar{Q}$	0.497 ± 0.029	0.833 ± 0.017
After drift 3	AUC	0.847 ± 0.036	0.900 ± 0.018
	$\bar{Q}$	0.469 ± 0.034	0.837 ± 0.024

5 Conclusions and Future Research Lines

It has been confirmed in Section 4 that it is indeed a good strategy to maintain highly diverse ensembles to obtain good classification performance shortly after the drift. Furthermore, the use of a bio-inspired solver such as HS is an proper way of building high diversity ensembles for batch learning scenarios where the evaluation of all possible ensembles of past learners at each time cannot be performed by an exhaustive method. When the time requirements or computational cost are stringent constraints, the HS algorithm allows reducing the number of improvisations and the size of the harmony memory, achieving a solution suitably balancing optimality and computational complexity under these conditions.

After the drift is detected, it is very critical to determine the time range over which a high diversity ensemble is convenient. This work has considered 10 time steps as a relevant interval for high diversity just to show the importance of a high diversity ensemble after the drift. However, this time period might change depending on the type of drift, its severity, the reliability of the drift detection and the statistics of the data considered in the problem. In general it is widely accepted that after a *large* number of time steps since the beginning of the drift, maintaining a high diversity becomes less important and even counterproductive with respect to low diversity ensembles. However, the exact quantification of this *large* number of time steps remains an open problem.

Diversity by itself is helpful to reduce the initial drop in accuracy that happens right after a drift, but not to provide convergence to the new concept. Although high diversity ensembles may help to cushion the initial increase in the error soon after the drift, they do not quickly adapt to the new concept (recovery from drifts). A practical workaround is to create a new ensemble after a drift detection. In this way, the technique would achieve the required equilibrium between *stability* and *plasticity* [21] to reduce the initial drop of accuracy after the drift while, at the same time, to be able to adapt to the new concept.

Also as a future research line it might be of interest to delve into the influence of the size of the ensemble in order to establish a mechanism to find the proper size at each point while simultaneously maintaining a certain level of diversity in

the ensemble. There is a trade-off between the severity degree of the disagreement among the members of the ensemble and the number of base learners within it. On the other hand, also the number of samples in the batch may affect this equilibrium, which will also be subject of further investigation in the future.

Acknowledgments

This work has been supported by the Basque Government through the ELKARTEK program (ref. KK-2015/0000080, BID3A project) and BID3ABI project.

References

1. L. Atzori, A. Iera, G. Morabito: The Internet of Things: A survey. Computer Networks, vol. 54(15), pp. 2787-2805 (2010)
2. L.L. Minku, A.P. White, X. Yao: The impact of diversity on online ensemble learning in the presence of concept drift. IEEE Transactions on Knowledge and Data Engineering, vol. 22(5), pp. 730-742 (2010)
3. L.L. Minku, X. Yao: DDD: A new ensemble approach for dealing with concept drift. IEEE Transactions on Knowledge and Data Engineering, vol. 24(4), pp. 619-633 (2012)
4. Z.W. Geem, J.H. Kim, G. Loganathan: A New Heuristic Optimization Algorithm: Harmony Search. Simulation, vol. 76(2), pp. 60-68 (2001)
5. V.R. Pandi, B.K. Panigrahi, S. Das, Z. Cui: Dynamic economic load dispatch with wind energy using modified harmony search. Int. J. Bio-Inspired Comput., vol. 2(3/4), pp. 282-289 (2010)
6. S. Salcedo-Sanz, A. Pastor-Sánchez, J. Del Ser, L. Prieto, Z. W. Geem: A Coral reefs optimization algorithm with harmony search operators for accurate wind speed prediction. Renewable Energy, vol. 75, pp. 93–101 (2015)
7. M.H. Scalabrin, R.S. Parpinelli, C.M. Bentez, H.S. Lopes: Population-based harmony search using GPU applied to protein structure predictionv. International Journal of Computational Science and Engineering, vol. 9(1/2), pp. 106 (2014)
8. R. Zhang, L. Hanzo: Iterative multiuser detection and channel decoding for DS-CDMA using Harmony Search. IEEE Signal Processing Letters, vol. 16(10), pp. 917-920 (2009)
9. D. Manjarres, J. Del Ser, S. Gil-Lopez, M. Vecchio, I. Landa-Torres, R. Lopez-Valcarce: A novel heuristic approach for distance- and connectivity-based multihop node localization in wireless sensor networks. Soft Computing, vol. 17(1), pp. 17-28 (2013)
10. D. Manjarres, I. Landa-Torres, S. Gil-Lopez, J. Del Ser, M.N. Bilbao, S. Salcedo-Sanz, Z.W. Geem: A survey on applications of the harmony search algorithm. Engineering Applications of Artificial Intelligence, vol. 26(8), pp. 1818-1831 (2013)
11. Z.W. Geem, C.L. Tseng, J.C. Williams: Harmony search algorithms for water and environmental systems. Music-Inspired Harmony Search Algorithm, pp. 113-127 (2009)
12. Z. Karimi, H. Abolhassani, H. Beigy: A new method of mining data streams using harmony search. Journal of Intelligent Information Systems, vol. 39(2), pp. 491-511 (2012)

13. M. N. Bilbao, J. Del Ser, S. Salcedo-Sanz, C. Casanova-Mateo: On the application of multi-objective harmony search heuristics to the predictive deployment of firefighting aircrafts: a realistic case study. International Journal of Bioinspired Computation, vol. 7, N. 5, pp. 270-284 (2015)
14. J. Žliobaitė, Indre;Pechenizkiy, Mykola;Gama: An overview of concept drift applications. Big Data Analysis: New Algorithms for a New Society, pp. 91-114 (2016)
15. J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, A. Bouchachia: A survey on concept drift adaptation. ACM Computing Surveys (CSUR), vol. 46(4), pp. 1-37 (2014)
16. G. Ditzler, R. Polikar, N. Chawla: An Incremental Learning Algorithm for Non-stationary Environments and Class Imbalance. International Conference on Pattern Recognition (ICPR), pp. 2997-3000 (2010)
17. T.G. Ditterrich: Machine Learning Research: Four Current Directions. Artificial Intelligence Magazine, vol. 4, pp. 97-136 (1997)
18. C.J. Kuncheva, L. I., and Whitaker: Measures of Diversity in Classifier Ensembles. Machine Learning, vol. 51(2), pp. 181-207 (2003)
19. G.U. Yule: On the association of attributes in statistics: with illustrations from the material of the childhood society, &c. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, vol. 194, pp. 257-319 (1900)
20. W.N. Street, Y. Kim: A streaming ensemble algorithm (SEA) for large-scale classification. ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 377-382 (2001)
21. S. Grossberg: Nonlinear neural networks: principles, mechanisms, and architectures. Neural Networks, vol. 1(1), pp. 17–61 (1988)