January - February 2020
ISSN: 0193 - 4120 Page No. 1807 - 1820

Evolving Ensemble Classifier for Mining Data Stream with Concept Drift

Nalini Nagendhiran[1,a] and Lakshmanan Kuppusamy[1]
[1]School of Computer Science and Engineering
VIT, Vellore, 632 014, India.
{nalini, klakshma}@vit.ac.in

Article Info
Volume 82
Page Number: 1807 - 1820
Publication Issue:
January-February 2020

Article History
Article Received: *14 March 2019*
Revised: *27 May 2019*
Accepted: *16 October 2019*
Publication: *07 January 2020*

Abstract
Learning concept drift is a challenging task in a non-stationary environment. Concept drift is concerned with learning from data, whose statistical data distribution changes over time. In recent days, ensemble classifiers have become a popular technique and more work has been carried out in data stream classification for non-stationary environment. Ensemble classifiers provide a natural way to adapt the changes which increase the classification accuracy than a single classifier. In this paper, we propose an Evolving Ensemble Classifier (EEC) based on ensemble classification technique which improves the performance of the learning model in the presence of concept drift. The proposed method EEC modifies the weighting function of Accuracy Updated Ensemble (AUE2e) algorithm. Our proposed algorithm EEC is compared with the existing well-known ensemble algorithms such as OzaBaggingo, AWEe and AUE2e on synthetic and real-world datasets. The experimental results show that the accuracy of the proposed algorithm EEC is substantially increased, regardless of the type of concept drift.

Keywords: *Data stream classification, Concept drift, Non-stationary environment, Ensemble learning, and online classifier.*

I. Introduction

Several data mining algorithms have been developed for stationary environment [1]. However, with the advent of technologies like Internet of Things, Wireless Sensor Networks, Machine to Machine communications, they result to huge amounts of data stream in non-stationary environments [2]. In dynamic environments, the data stream may evolve and its statistical characteristics can change over a period of time [3]. This leads to the domain of concept drift [3][4]. Concept drift can be defined as unforeseeable changes over time in the underlying distribution of data stream. This can be a gradual change or an abrupt change over time and is difficult to identify the change.

The evolving nature of the data stream increases the need of memory and computational power. Therefore, classifiers that deal with concept drifts are forced to implement forgetting, adaptation, or drift detection mechanisms in order to adjust to non-stationary environments. For example, customer purchase patterns can be influenced by many factors like economic conditions, trends, age and introduction of new products in the market, etc.

Concept drift reduces the performance of the learning algorithms due to the changes in data distribution over time. Therefore, the learning algorithm has to be modified accordingly to incorporate the concept drift in data stream and to make the analysis more precise and relevant. Ensemble-based learning method [5] provides a natural way to adapt the changes because of their modularity characteristics. It combines results from multiple classifiers through different techniques like averaging, mode, etc. This enhances the accuracy and scalability of the model than a single classifier [5][6].

In a non-stationary environment, Ensemble approaches can be classified into block-based approach [7-12] and online based approach [13-16]. In the latter, the data is processed continuously, whereas, the block-based approach is processed in equal-sized data blocks.

The block-based technique [7-12] divides into block of instances and then check the classifier model on the newly arrived instances. Based on the accuracy of the classifier, the least accuracy classifier is removed from the ensemble and is replaced by a new classifier that keeps the model up to date. The disadvantage of using this approach is that one cannot respond immediately to concept drift. Moreover, defining the size of the block of instances are problematic. Larger blocks of data for this training would produce better classifiers, but it might incorporate more than one concept drift. Smaller blocks highlight every drift but produce classifiers with less accuracy. Thus it is important to define a trade-off between the two and to develop an effective block-based ensemble model for stream data classification.

In online approach [13-16], on every incoming example, the classifier incrementally updates the existing model rather than building a new one. There are no blocks in this approach and it responds faster to sudden drift. The disadvantage of using this approach is requirement of huge computing power to verify the class labels in each incoming example. The classifier components, unlike block-based approach, are not updated periodically and need a special mechanism to replace older and weaker classifier.

We propose an approach with a combination of block-based model and incremental online approach to get better results. The proposed method EEC is the process of learning a number of ensemble classifiers, and combining them to predict incoming data using new weighing mechanism. The new weighting function is used to identify the best classifier which correctly classifies the newly incoming instances. Thus, EEC performs continuously adapt to the changes (in other words drifts) which improves the performance of the learning model without using an explicit drift detection mechanism.

The remainder of the paper is structured as follows. Section 2 reviews the various ensemble approaches and identify the limitation of existing approaches. Section 3 proposes the new algorithm Evolving Ensemble Classifier (EEC). Section 4 compares EEC with block-based ensemble

January - February 2020
ISSN: 0193 - 4120 Page No. 1807 - 1820

approaches namely Accuracy Weighted Ensemble (AWE_e), Accuracy Updated Ensemble ($AUE2_e$) and the online based approach - $OzaBagging_o$ with synthetic and real-world data stream. Section 5 concludes the paper with some possible future works.

I ENSEMBLE APPROACHES IN EVOLVING DATA STREAM

Before, we discuss the various ensemble approaches, we first discuss the background of concept drift.

1 Concept Drift

The data stream (d_s) can appear as {X, y}, where X is a instance vector consisting of q attributes i.e., $X = (a_1, a_2, \ldots, a_q)$, and X is classified with a class labely $\in\{y_1, y_2, . . . ,y_m\}$. In non-stationary environments, data stream may evolving over time is called "Concept drift" and is defined as

$P_t (X, y_r) \neq P_{t+j}(X, y_r)$,

where P_t and P_{t+j}, denote the data joint distribution at time t and t+j ($j\geq1$) respectively. More specifically, the classifier classifies the instance X and is predicted with class label as y_r, at time 't'. Later at time 't+j', data distribution may change (suddenly/gradually) and when instance Xrepeats but may not be predicted as class label y_r. Generally, concept drift patterns are classified into three form of drifts that are based on speed of changes happen in the data distribution (see Fig.1).

- Gradual drift: A new concept slowly changes over a period of time and replaces the existing one.
- Sudden drift: A new concept takes place in a short period of time.
- Recurrent drift: An old concept could reoccur over a period of time.

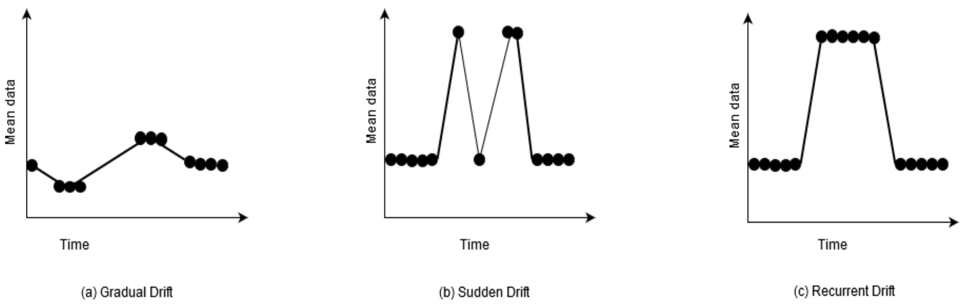


Fig. 1. Types of Concept Drift Pattern

January - February 2020
ISSN: 0193 - 4120 Page No. 1807 - 1820

2 Ensemble Approaches for Non-stationary data stream

In this section, we present the survey on various ensemble classifiers [7-16] on the block-based and online ensemble methods which deal with evolving data streams.

One of the first block-based approaches is the Streaming Ensemble Algorithm (SEA) [7]. It focuses on an ensemble model that reads blocks of data at a time which reduces the memory requirement of the model. A new classifier is added to the ensemble if it enhances the performance of the model. The observations are made that increasing the number of classifiers will result to enhance the accuracy of ensemble classifier; however, when the number of classifier increases, the computational complexity also increases drastically. The unique factor of this technique is that the replacement of a classifier is not just based on accuracy, but is also based on the classification diversity.

Accuracy Weighted Ensemble AWE_e[8] is a technique proposed by Wang at el. [8]. In this approach, the data stream (d_s) can be partitioned into equal sequential block b_1, b_2, b_3,..,bn each block has s instances. Every incoming block b_i, the error rate of component classifier $C_j \in E_k$ is estimated by following Equation (1) and Equation (2).

$$MSE_{ij} = \frac{1}{|b_i|} \sum_{\{x,y\} \in b_i} \left(1 - f_y^j(x)\right)^2 \qquad (1)$$

$$MSE_r = \sum_y p(y) * \left(1 - p(y)\right) \qquad (2)$$

where function $f_y^j(x)$ is the probability given by classifier C_j that x is an instance with class y. The value of Mean Square Error (MSE_{ij}) is the prediction error rate of component classifiers (Cj) on recent block, b_i and MSE_r, Mean Square Error of randomly predicting classifier and is used as a reference point to predict based on the current class distribution. The weight has been calculated based on performance of the classifier using Equation (3).

$$WT_{ij} = MSE_r - MSE_{ij} \qquad (3)$$

This model does not deal with abrupt concept drifts well. In such cases, it discards the entire ensemble and starts developing it from scratch. This utilizes a lot of resources unnecessarily. The performance of both the above techniques depend on the incoming block size. This high dependency on the block size can be disadvantage to the model. Another approach of this family is Accuracy Updated Ensemble ($AUE2_e$) [9][10]. This approach is extended version of AWE_e and it calculated the error of classifier as mentioned in equation 1 and equation 2. But, $AUE2_e$ have used updated weighting function of AWE (see Equation (4)).

$$WT_{ij} = \frac{1}{MSE\ r + MSE\ ij + \epsilon} \qquad (4)$$

The very small constantϵ (is assumed to have the value 0.01 in our paper and as well in referenced papers) is used in the weighting function in order to avoid exceptional conditions (i.e., denominator should not be zero when MSE_{ij} = 0 = MSE_r).In

$AUE2_e$, the new set of classifiers that are formed after each incoming examples are updated and weighted based on their accuracy. This is a part of the incremental approach which is incorporated in the $AUE2_e$ technique.

Another family of block-based approach is Learn ++ NSE [11]. This method handles sudden drift well than SEA. It invokes previous classifiers whenever needed and disables them when they are not relevant to new data. Thus, this ensemble technique is used efficiently for extracting knowledge from the instances. Another technique for particularly dealing with sudden drifts is Batch Weight Ensemble (BWE) [12]. It includes a change detector in the ensemble to incorporate changes. Unlike AWE_e, BWE could not construct all new classifiers for a new batch and the approach helps to save memory. The batch drift detection model creates a table and saves the cumulative accuracy of the different classifiers. This helps in showing the data trends across the batches using a linear regression model. When continuously drifting trends in the same direction are observed, this batch model gives out a warning signal, a new classifier is added. Then weights are given to the classifiers which help in removing the weaker classifiers.

The Weighted Majority Algorithm [13] learns as it grows and as it gets more and more classification examples. This algorithm requires a large training dataset for learning and improves the model. This can be a disadvantage in the case of small dataset.

Online bagging and boosting [14] is an online ensemble classifier. It has a low overheads due to the online approach and updates the model incrementally. Each example is presented to a component k times, where k is a constant determined by the Poisson distribution. The performance of this technique is comparable to its batch counterparts. Dynamic Weighted Approach [15] is another online ensemble technique. The disadvantage of using this technique is that when large datasets are involved, it generates a large number of classifiers or components. Thus pruning the classifiers becomes an important additional step while using this technique. The Leverage Bagging approach [16] adds more randomization to the component classifier which enhances the accuracy and diversity of the learning model. Although Leveraging Bagging is faster in processing the instances, but, its accuracy is lesser than other approaches.

II EVOLVING ENSEMBLE CLASSIFIER APPROACH

In this section, we discuss the framework of the proposed method, the Evolving Ensemble Classifier (EEC). The existing ensemble classifiers are able to handle only one type of drift. In real-world scenarios, data stream arrives in any type of drift which is unable to identify in advance. The main goal of the proposed system EEC is to design a new ensemble-based classifier that quickly adapts to various types of concept drifts such as gradual, sudden and recurrent drift.

The EEC divides the instances into equal size of blocks, and apply the new component classifier to each instance in the block. In general, we can use

January - February 2020
ISSN: 0193 - 4120 Page No. 1807 - 1820

any online learning algorithms as a component classifier. The EEC uses Hoeffding Tree [17] as a component classifier because of its incremental nature. The performance of the component classifiers is assessed by prediction error rate on newly arrived data block and accordingly, weight is assigned to the classifier. The EEC proposes a new weighting function which is derived from Wang at el [8]. They proved that weighted ensemble classifiers reduce prediction error rates than a single classifier. The classifier can be added or removed from the ensemble based on its weight. Figure 2 summarizes the overall execution framework of the proposed method EEC.

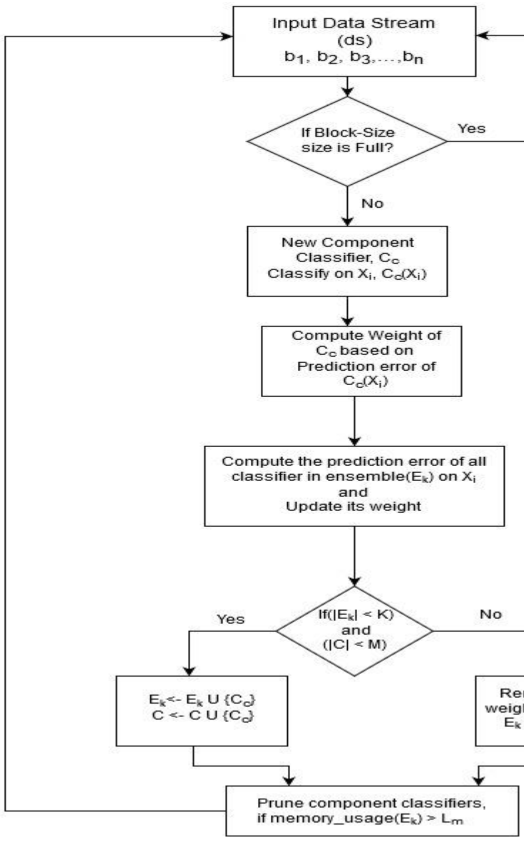


Fig. 2. Framework for Proposed method EEC

```
Algorithm: 1 Evolving Ensemble Classifier (EEC)
Input:
ds: data stream of instances split into blocks {b1, b2, b3,…,bn}.
    B: size of block =1000
    K: number of ensemble classifiers =10
    M: number of component classifiers =30
      Lm: memory limit (1000KB)
Output: Ek: ensemble of K incrementally classifier with weights
```

January - February 2020
ISSN: 0193 - 4120 Page No. 1807 - 1820

```
Eₖ = Ø
C = Ø
for all data blocks bᵢ ∈ dₛ
classify new component classifier C_c on bᵢ
  compute the error of C_c using equation (1)
  calculate weight WTᵢⱼfor C_c using (3)
  // incrementally train classifier Cᵢ with bᵢ
  for each classifier in Cᵢ ∈ C do
    apply Cᵢ on bᵢ to compute error(MSEᵢⱼ)using equation (1)
 calculate weight WTᵢⱼfor Cᵢ using (3)
  end for
  if | Eₖ | < K and |C| < M then
Eₖ = Eₖ U { C_c }
 C = C U { C_c }
 else
    remove minimum weighted classifier from ensemble(Eₖ)and component
    classifier(C) by C_c
  end if
  if memory_ usage(Eₖ) > Lₘ then
    Prune(decrease size of) component classifier in C
  end if
 Eₖ = top-K weighted classifiers in the ensemble
end for
```

Algorithm 1 EEC assumes the data stream (d_s) can be partitioned into equal sequential block b_1, b_2, b_3,..,bn each block has s instances. Every incoming block b_i, the error rate of component classifier $C_c \in E_k$ is estimated by using Equation (1) and Equation (2).

$$WT_{ij} = \frac{\text{MSE r}}{(\text{MSE ij} + \epsilon)} \quad (5)$$

This (Equation (5)) is the new weighting function (WT_{ij}) for EEC is mentioned. The WT_{ij} is proportional to the $MSE_{ij,}$ but it is reversely proportional to MSE_r.

The weighted component classifier (C_c) is added into the ensemble (E_k) and C if the number of the ensemble is less than K and M. Otherwise, the poorest performing classifier (least weighted classifier, C_p) is eliminated from the ensemble (E_k) and C by C_c. If the memory usage of the ensemble is greater than the memory limit which is defined in EEC, then to decrease the size of the ensembles, the least active leaves of component in HT will be removed. Finally, the ensemble (E_k) is prepared to classify the next incoming block.

TEST Engineering & Management
January - February 2020
ISSN: 0193 - 4120 Page No. 1807 - 1820

III EXPERIMENTAL EVALUATION

In this section, we discuss the results of various experiments done on EEC with other ensemble approaches.

The proposed method of EEC is implemented in Python. All the experiments are performed on Windows OS-Pro 64-bit Intel Core i5 7200 CPU @2.50 GHz with 16GB of RAM. The EEC uses HT enhanced with adaptive Naive Bayes leaf prediction as a base classifier. The HT tree properly branches out and reflects the changes in the data stream. We fix the parameters of HT as grace period = 200 (instances),

memory limit (L_m) = 1000KB.

We compare EEC with $OzaBagging_o$, AWE_e and $AUE2_e$ with different types of drifted data stream. These datasets could be synthetic and real-world. The prequential accuracy [3][18] is a very noble metric to assess the classifier efficiency in a non-stationary environment with the presence of concept drift. This metric is tested on each instance and then applied for training. Therefore, the accuracy is updated incrementally with maximum available data. As a result, the classifier is continuously tested on unseen instances which helps to improve the performance of the new model.

1 Benchmarking Datasets

In this section, we provide information about synthetic and real-world datasets as shown in Table 1.

Synthetic dataset

In our experiment, all the synthetic datasets are generated using the Python Stream Generator Package.

- **SEA**

SEA was implemented by Street and Kim [7]. In this paper, we generate 50000 instances with 3 attributes and 1 target. The class decision boundary is decided by $y_1 + y_2 \leq \alpha$, where y_1 *and* y_2 attributes are relevant for prediction and α is a threshold rate. Target class labels depend upon four functions f1($\alpha = 8$), f2($\alpha = 9$), f3($\alpha = 7$), and f4 ($\alpha = 9.5$). It induces gradual drift from f1 to f4 in

SEA_G for 10000 instances (from 35000 to 45000 instances). We also induce sudden drift by suddenly shifting from f1 to f4 in SEA_S at the arrival of 30000th instances.

- **Rotating Hyperplane:**

The rotating hyperplane [19]is represented in k-dimensional space by $\sum_{i=1}^{k} x_i s_i = x_0$ where s_i are instances, $x_i$is the corresponding weight of each attribute and x_0 is the total weight. The instances which satisfy the condition, $\sum_{i=1}^{k} x_i s_i \geq x_0$, labeled as 1, otherwise 0. In this paper, a hyperplane is used to simulate the gradual drift by rotating the hyperplane slightly with each consecutive instance. The presence of gradual drift for 3000 instances (from 4000 to 7000 instances) are generated from 10000 instances

January - February 2020
ISSN: 0193 - 4120 Page No. 1807 - 1820

with 2 attributes and 1 target class as known as $HYPERPLANE_G$.

- **Agarwal Generator**

The Agarwal generator [20] used in the proposed EEC to generate 50000 instances with 9 attributes and 1 target class. This generator is about processing loans and predicts the class labels by ten functions whether the loan can be approved or not. It persuades gradual drift in AWG_G for 10000 instances (from 35000 to 45000 instances).

Real-world datasets

There are four real-world datasets that are used in the experiment. Basically, real-world datasets are unable to identify the type of concept drift in advance. The proposed method EEC is certainly evolving with dynamic data stream and quickly adapt to changes (see Table 4).

- **Weather**

The Weather dataset [21] is a normalized version of the NOAA dataset (50 years of weather data) from a post at Bellevue, Nebraska's Offutt Air Force Base, USA. A number of features such as temperature, pressure, visibility, wind speed, etc.are measurements taken regularly. The number of feature vectors was reduced to eight and the task of classification was to predict whether on a particular day there was rain or not.

Table 1 Properties of dataset

Dataset	Number of instances	Number of attributes	Number of Classes	Drift pattern
$HYPERPLANE_G$	10000	2	2	Gradual
SEA_S	50000	3	2	Sudden
SEA_S	50000	3	2	Sudden
AWG_G	50000	9	2	Gradual
Weather	18160	8	2	Unknown
Click-Prediction	39949	12	2	Unknown
Electricity	45312	8	2	Unknown
GAS_R	13910	128	6	Recurrent (manual induced)*

* Manually, we rearranged the dataset to have a recurring drift, namely GAS_R.

- **Click-Prediction**

The Click-prediction dataset obtained by the 2012 KDD Cup [22]. The dataset has the advertising information displayed in a search engine consisting of search results and ads which used to predict whether the user is clicking on the ads or not.

- **Electricity**

The Electricity market dataset defined by M Harries [23] and Gama [4] is another commonly used dataset. The data was collected from the Australian New South Wales Electricity Market, where the electricity prices are non-stationary and are influenced by the market supply and demand.

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 1807 - 1820

• Gas-Drift

The Gas-Drift dataset [24] is prepared by Alexander Vergara which gives information about the degradation of sensors. The dataset was collected in a gas delivery platform located at the ChemoSignals Laboratory at the BioCircuits Institute, University of California San Diego, from January 2007 to February 2011. We arranged the dataset to have a recurring drift, namely GAS_R. This data reorganization was done to ensure that a sufficient number of experiments in each class and month were distributed as recurrently as possible when training the classifier.

2 Impact of Ensemble and Component Classifiers on EEC

In this section, we experiment with EEC with a different number of component classifiers (M) and ensembles (K). Table 2 shows the impact of M and K on the performance of the EEC according to various drift scenarios. It is noted that increasing the size of M and K have a very minimal impact on EEC performance with synthetic and real-world dataset. Though M=100 and K=25 gives better results, we have chosen M=30 and K = 10 to reduce the computational complexity of ensemble classifiers that also gives better accuracy rate for EEC on the average cases (see Table2).

3 Analysis of Accuracy on EEC

In this section, we examine the results of EEC accuracy on both synthetic and real-world datasets with other ensemble classifiers, namely $OzaBagging_o$, AWE_e, and $AUE2_e$.

As shown in Table 3 and Table 4, the proposed method EEC has achieved a better result when compared to $OzaBagging_o$, AWE_e and $AUE2_e$. Table 3 proves that the EEC gives more accuracy on $HYPERPLANE_G$ (4%) and SEA_G (2%) with a gradual drift dataset. For the SEA_S dataset, EEC has given 8% of accuracy of 8% with $AUE2_e$ and 2% with AWE_e but nearly same as $OzaBagging_o$. For the AWG_G dataset, online ensemble $OzaBagging_o$ performs 2% (overall) better than other approaches.

Table. 2. Accuracy of EEC with different number of M and K

	SEA_G	SEA_S	$HYPERPLANE_G$	AWG_G	Weather	Click-Prediction	Electricity	GAS_R
M=10 & K=3	91.18	95.82	88.08	94.41	75.33	84.11	78.92	60.56
M=30 & K=10	91.31	96.01	91.22	93.92	76.23	84.25	79.87	61.32

January - February 2020
ISSN: 0193 - 4120 Page No. 1807 - 1820

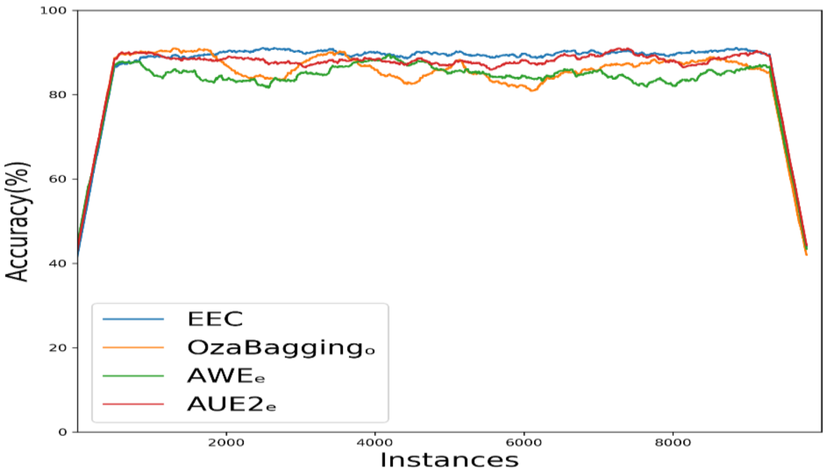
M=50 & K= 15	91.28	95.65	91.28	91.16	76.61	84.41	79.91	61.45
M=100 & K=25	91.10	95.51	92.01	91.71	76.75	84.92	79.95	61.85

Table. 3. Accuracy of EEC with other ensemble approaches in synthetic datasets

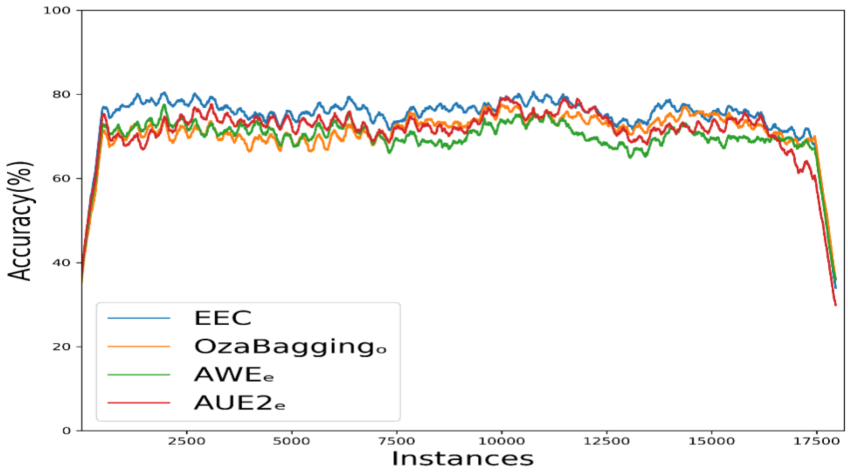
Ensemble Approaches	Synthetic Datasets			
	SEA_S	SEA_G	$HYPERPLANE_G$	AWG_G
	50K	50k	10K	50K
EEC(Proposed Method)	96.01	91.31	91.22	93.92
$OzaBagging_o$	98.66	88.80	81.18	95.22
AWE_e	94.89	85.89	84.43	90.33
$AUE2_e$	88.21	89.21	86.28	93.51

Table. 4. Accuracy of EEC with other ensemble approaches in real-world datasets

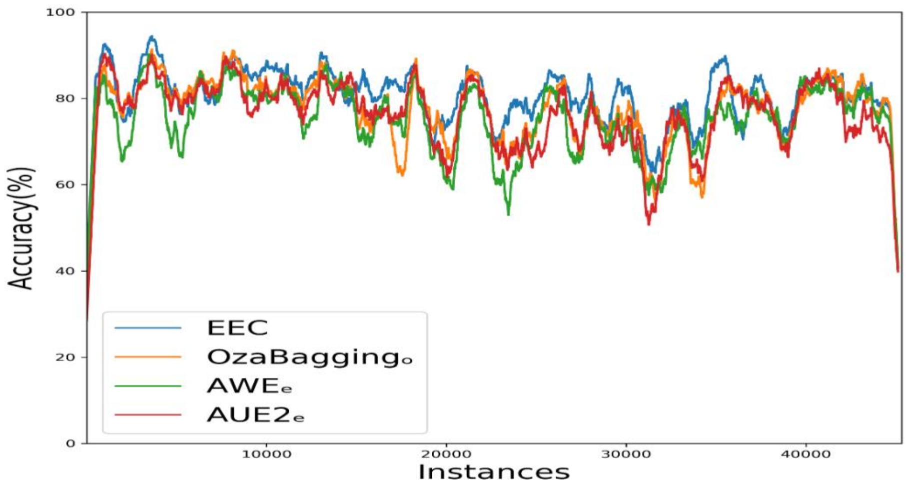
Ensemble Approaches	Real-World Datasets			
	WEATHER 50K	GAS-DRIFT 10K	ELECTRICITY 45K	CLICK-PREDICTION 39K
EEC(Proposed Method)	76.23	61.32	79.87	84.25
$OzaBagging_o$	73.19	54.69	77.25	80.03
AWE_e	71.45	48.01	73.26	78.03
$AUE2_e$	74.12	56.76	76.54	82.01



3a.$HYPERPLANE_G$ dataset



3b. Weather dataset



3c. Electricity dataset

Fig. 3 Accuracy of synthetic and real-world datasets with EEC, OzaBagging_o, AWE_e and AUE2_e

Table 4 depicts that the EEC has given more accuracy than other block-based and online ensemble approaches. The EEC has given an additional 2% accuracy for Weather and Click-prediction datasets compared to other ensemble approaches. For the electricity dataset, EEC has given increased accuracy of 6% in AWE_e, 3% in AUE2_e and 2% in OzaBagging_o online ensemble classifier. The EEC provides 12% more accuracy than AWE_e and also performs better than other approaches for the GAS_R dataset which induced by recurrent concept drift. Hence, it proves that the EEC performs well on recurrent drift and gradual drift, but moderately on sudden drift without using explicit drift detection mechanism.

Fig.3a, Fig.3b, and Fig.3cshows the comparative analysis graph for accuracy using HYPERPLANE_G, Weather and Electricity dataset on EEC with OzaBagging_o, AWE_e and AUE2_e.

Thus, the experimental evaluations on synthetic and real-world datasets show that our proposed algorithm EEC adapt to the different types of drift pattern such as sudden, gradual, and recurrent. The adaptiveness of EEC helps to improve the classifier accuracy in the non-stationary environment.

IV Conclusion

In this paper, we have presented a detailed ensemble approaches for data stream classification and emphasized the benefits and boundaries of the approaches. Further, we proposed Evolving Ensemble Classifier (EEC) and achieved better accuracy. The main idea behind the EEC uses a new weighting function mechanism to improve the performance of the classifier. The proposed method EEC handles the different types of drift patterns such as sudden, gradual, and recurrent in data stream. The obtained results by EEC method is compared with similar types of ensemble approaches and from the comparison it can be seen that EEC method performs overall very well.

Current research direction on dimensionality reduction methods for high-dimensional streaming data is inadequate [25]. Therefore one can plan to work on dimensionality reduction [26] in order to improve the quality and efficiency of classifier. Also, one can aim to investigate the proposed method EEC with unlabeled data stream [27].

References

[1] Tom Mitchell. Machine Learning. McGraw Hill, 1997.

[2] G Ditzler, M Roveri, C Alippi, and R Polikar. Learning in nonstationary environments: a survey.*IEEE Comput. Intell. Mag.*2015; *10*(4), 12-13.

[3] Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine learning*, *23*(1), 69-101.

[4] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, *46*(4), 44.

[5] Dietterich, T. G. (2000, June). Ensemble methods in machine learning. In *International workshop on multiple classifier systems* (pp. 1-15). Springer, Berlin, Heidelberg.

[6] D.H. Wolpert , The supervised learning no-free-lunch theorems, in: Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications, 2001, pp. 25–42 .

[7] W. Nick Street and Yong Seog Kim. 2001. A streaming ensemble algorithm (SEA) for large-scale classification. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '01). ACM, New York, NY, USA, 377-382. DOI=http://dx.doi.org/10.1145/502512.502568.

[8] Wang, H., Fan, W., Yu, P. S., & Han, J. (2003, August). Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 226-235).

[9] Brzeziński, D., & Stefanowski, J. (2011, May). Accuracy updated ensemble for data streams with concept drift. In *International conference on hybrid artificial intelligence systems* (pp. 155-163). Springer, Berlin, Heidelberg.

[10] Brzezinski, D., & Stefanowski, J. (2013). Reacting to different types of concept drift:

The accuracy updated ensemble algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(1), 81-94.

[11] R. Elwell , R. Polikar , Incremental learning of concept drift in nonstationary environments, IEEE Trans. Neural Netw. 22 (10) (2011) 1517–1531.

[12] M. Deckert , Batch weighted ensemble for mining data streams with concept drift, in: International Symposium on Methodologies for Intelligent Systems, Springer Berlin Heidelberg, 2011, pp. 290–299.

[13] Littlestone, N., & Warmuth, M. K. (1994). The weighted majority algorithm. *Information and computation*, *108*(2), 212-261.

[14] N C Oza. Online bagging and boosting. *In:*2005 IEEE international conference on systems, man and cybernetics, Waikoloa, HI, USA.2005, Vol. 3, p. 2340-5.

[15] Kolter, J. Z., & Maloof, M. A. (2007). Dynamic weighted majority: An ensemble method for drifting concepts. *Journal of Machine Learning Research*, *8*(Dec), 2755-2790.

[16] A Bifet, G Holmes and B Pfahringer. Leveraging bagging for evolving data streams.*In:*Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Barcelona, Spain. 2010, p. 135-15.

[17] Domingos, P., & Hulten, G. (2000, August). Mining high-speed data streams. In *Kdd* (Vol. 2, p. 4).

[18] Dawid, A. P. (1984). Statistical theory: The prequential approach. Journal of the Royal Statistical Society-A 147, 278-292.

[19] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In KDD'01, pages 97–106, San Francisco, CA, 2001. ACM Press.

[20] Rakesh Agrawal, Tomasz Imielinksi, and Arun Swami. "Database Mining: A Performance Perspective", IEEE Transactions on Knowledge and Data Engineering, 5(6), December 1993.

[21] Ueno, K., Fujii, H., Yamada, H., & LIU, L. (2001). Federal Climate Complex Global Surface Summary of Day Data, VERSION 5. 1999. *Journal of the Meteorological Society of Japan. Ser. II*, *79*(1), 419-434.

[22] https://www.openml.org/d/1218

[23] Harries, M., & Wales, N. S. (1999). Splice-2 comparative evaluation: Electricity pricing.

[24] Alexander Vergara, Shankar Vembu, Tuba Ayhan, Margaret A. Ryan, Margie L. Homer, and Ramón Huerta, Chemical gas sensor drift compensation using classifier ensembles, Sensors and Actuators B: Chemical (2012) doi: 10.1016/j.snb.2012.01.074.

[25] Park, C. H. (2016). Dimension Reduction Methods on High Dimensional Streaming Data with Concept Drift. *KIPS Transactions on Software and Data Engineering*, *5*(8), 361-368.

[26] Qahtan, A. A., Alharbi, B., Wang, S., & Zhang, X. (2015, August). A pca-based change detection framework for multidimensional data streams: Change detection in multidimensional data streams. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 935-944). ACM.

[27] Ahmad, S., Lavin, A., Purdy, S., & Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, *262*, 134-147.