Adaptive Rotation Forests: Decision Tree Ensembles for Sequential Learning

Yu Sugawara[1] and Satoshi Oyama[2] and Masahito Kurihara[3]

***Abstract*— We have developed an ensemble-based approach for online machine learning: adaptive rotation forest and ADWIN adaptive rotation forest. We focused on rotation forest, an offline supervised ensemble algorithm with a particularly high prediction accuracy while all the features are continuous. Our objective was to develop a high-performance online ensemble method that uses a process similar to that of rotation forest in an online environment. Our experiments demonstrated that the proposed approach simplifies the tree structure used for the base learners, reduces memory consumption, and improves prediction accuracy for some data streams.**

I. Introduction

Data stream mining has become an emerging research topic in the data mining field. It has more requirements than traditional data mining because the input is a stream of data. Bifet et al. [3] summarized the most important requirements:

- Process an instance at a time, and inspect it (at most) once.
- Use a limited amount of time to process each instance.
- Use a limited amount of memory.
- Be ready to give an answer (prediction) at any time.
- Adapt to temporal changes.

These requirements make it difficult to apply ordinary supervised learning algorithms to data stream mining.

We have addressed the handling of online classification tasks, i.e. classification problems, with data streams as input. Our goal was to develop a method for accomplishing this task accurately and efficiently while satisfying the above requirements.

The importance of data stream mining algorithms is increasing. The amount of data available has been growing exponentially due to the emergence of the big data phenomenon [8], [16]. To handle such an enormous amount of data, we need a fast and efficient method that works in real time and with a reasonable amount of resources [3].

There are various data stream classification algorithms. The Hoeffding tree algorithm [7] constructs a decision tree in a sequential manner. Each node of the tree is split sequentially when enough samples have been accumulated to ensure a sufficient level of confidence. Online bagging [14] executes bagging sequentially. By statistically emulating bootstrap sampling, which is a vital element of bagging, we can approximate the effect of bagging even in an online environment. As shown above, reproducing powerful tools for regular supervised learning tasks in an online environment is one of the best practices for data stream classification.

[1]Hokkaido University yu.sugawara.1116@gmail.com
[2]Hokkaido University oyama@ist.hokudai.ac.jp
[3]Hokkaido University kurihara@ist.hokudai.ac.jp

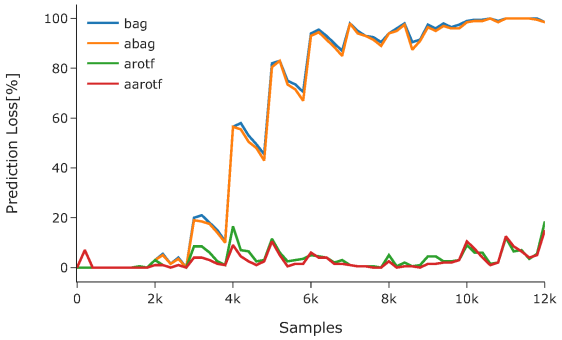

Fig. 1. Prediction accuracy for a portion of 4CRE-V1 data stream. Accuracy with existing methods, online bagging and adaptive window (ADWIN) bagging, declined because of concept drift while proposed methods, adaptive rotation forest (ARotF) and ADWIN adaptive rotation forest (AARotF), maintained high prediction accuracy.

We focused on rotation forest (RotF) [17], which has almost the same structure as bagging. One difference is that before the samples are passed to each base learner, principal component analysis (PCA) is executed on the samples to create a sparse rotation matrix that is unique to each base learner. Each of the base learners receives a sample after it is mapped by PCA. This pre-processing improves the performance and diversity of the base learners, both of which are essential for increasing the overall prediction accuracy in ensemble learning [9].

According to Bagnall et al. [1], RotF is the best classifier for problems with continuous features in comparison with support vector machine, random forest, XGBoost, and multilayer perceptron. This is our motivation: If we could emulate RotF in an online environment, it would be a powerful data stream classification algorithm.

Similar to our work, Pham et al. [15] and Heusinger et al. [10] reported that they improved the accuracy of ensemble prediction or reduced the execution time by giving different mappings to different base learners in an online environment. Both of these papers discuss implementing random projection [11] techniques in an online environment. The latter focuses on non-stationarity and reports that random projection does not adversely affect prediction accuracy and reduces execution time for data streams with concept drift.

However, the focus of these papers is on random projection, and the elements of RotF, i.e., learning rotation by PCA, have not been sufficiently discussed. We addressed this deficiency in our study and created the adaptive rotation forest

(ARotF) method for learning an ensemble of Hoeffding trees with rotation. The basic idea of ARotF is to replace each of the three components of RotF with an online algorithm:

- Replace the decision tree with a Hoeffding tree [7].
- Replace bagging with online bagging [14] or ADWIN bagging [5].
- Replace PCA with incremental PCA [18].

We performed comparative experiments on multiple data streams to evaluate the performance of the proposed approach. The results show that for data streams where PCA is effective, the proposed approach reduces the effect of concept drift and simplifies the structure of the tree. As a result, we observed an improvement in prediction accuracy and memory consumption. Fig. 1 shows example results.

After defining the problem in section II, we describe several existing online and/or supervised learning methods related to RotF in section III. In section IV, we present the detailed RotF procedure. In section V, we report the results of our comparative experiments. Finally, in section VI we discuss the benefits of RotF and the characteristics of the data streams for which it is effective.

II. Problem Definition

Let $\boldsymbol{x} = [x_1, \ldots, x_n]$ be a data point described by n features, and let y be a class label for the data. Our goal is to build an estimator that predicts y with high accuracy given $\boldsymbol{x}$.

In a data stream mining task, samples for training and prediction are given sequentially from the data stream one by one. The data stream is assumed to be non-stationary; i.e., it has concept drift. Therefore, we must consider the possibility that a highly accurate predictor at one moment may be obsolete the next.

III. Related Work

A. Hoeffding Tree

The Hoeffding tree [7] is a very fast incremental decision tree induction algorithm that can be trained from large stationary data streams. The algorithm builds a decision tree from the root to the end, similar to the C4.5 algorithm, an offline decision tree construction method. Since C4.5 cannot start construction until all data are available, it does not meet the requirements for data stream mining. In contrast, the Hoeffding tree algorithm builds a tree sequentially using only a small number of samples without waiting for all the data to be received. The advantage of each potential node split is calculated, and the best one is executed only if its advantage exceeds a threshold, which is mainly determined by the number of samples the node has observed.

The Hoeffding tree algorithm has a hyperparameter (δ) that guarantees the reliability of each split. That is, there is a theoretical guarantee if a sufficiently low δ is specified by the user. A decision tree constructed by the algorithm is nearly identical to one constructed when all the training data in the data stream can be randomly accessed.

Since the Hoeffding tree algorithm is not new, a number of derivatives have been proposed. However, we used Hoeffding trees as base learners in our study because of their relatively low computational cost and high prediction accuracy and the existence of a theoretical guarantee for the reliability of the constructed decision tree.

B. Online Bagging / ADWIN Bagging

Similar to dataset mining, an ensemble of decision trees can improve prediction accuracy in data stream mining. In this paper, we employ online bagging [14] and ADWIN bagging [5] as ensemble methods in an online environment.

Online bagging is an ensemble method for data stream input. For normal bagging, the probability of a given original sample being included in the bootstrap sample k times can be expressed as a binomial distribution,

$$P(K = k) = \binom{N}{k} \left(\frac{1}{N}\right)^k \left(1 - \frac{1}{N}\right)^{N-k},$$

where N is the size of the dataset. When N is sufficiently large, this distribution can be approximated with a Poisson distribution:

$$P(K = k) \sim \mathrm{Poisson}(1).$$

Thus, the same effect as bagging in an online environment can be obtained by sampling k from the Poisson distribution for each base learner and feeding the sample to each learner k times.

In ADWIN bagging, an extension of online bagging, the ensemble method itself actively responds to concept drift. Therefore, concept drift can be dealt with even if the base learner does not support non-stationary data streams. The prediction accuracy of each base learner is managed using an adaptive window [4], which is a sliding window with the ability to adaptively determine the window length. Whenever a new sample $(\boldsymbol{x}, y)$ arrives, ADWIN bagging evaluates the base learner's prediction $\hat{y}$ and records sequentially whether it is equal to the true class label y in the corresponding ADWIN. If the prediction accuracy of the base learner has deteriorated substantially, ADWIN changes the window length, meaning that the current base learner has become obsolete due to concept drift. ADWIN bagging thus resets the base learner with the worst prediction accuracy at that time. This operation maintains the performance of the entire ensemble by discarding learners that are not keeping up with the concept drift.

C. Incremental PCA

PCA is a commonly used method for reducing the dimension of data without losing most of the variations in data points. Incremental PCA (IPCA) [18] is an extension for approximating the results of conventional PCA in an online environment.

There are several methods for executing PCA in a sequential manner. So why did we choose IPCA?

First, IPCA works relatively fast and has a sufficiently high eigenvalue prediction accuracy. Cardot et al. [6] conducted experiments showing that IPCA and candid covariance-free incremental PCA provide a very good compromise between statistical accuracy and computation time compared with

Algorithm 1 (ADWIN) Adaptive Rotation Forest

Input: S: Data Stream, L: Ensemble size, f: Number of features per subset

Initialize L Hoeffding Trees $\boldsymbol{F} =< F_1, \ldots, F_L >$.
for $i \leftarrow 1 \ldots L$ **do**
  Divide the features into subsets $< S_1, \ldots S_N >$, where $N = m/f$.
  for $j \leftarrow 1 \ldots N$ **do**
    Initialize IPCA $\boldsymbol{I}_{ij}$ responsible for the rotation of the feature subset S_j of the Hoeffding Tree F_i.
  end for
end for
while Data stream S continues **do**
  if Learn from data$(\boldsymbol{x}, y)$ **then**
    for $i \leftarrow 1 \ldots L$ **do**
      *Input the mapped sample $(R\boldsymbol{x}, y)$ into the Hoeffding tree F_i to obtain the prediction $\hat{y}_i$.
      *If any ADWIN detects a deterioration in prediction accuracy, reset worst performing base learner.
      Sample p from $Poisson(1)$.
      for $j \leftarrow 1 \ldots p$ **do**
        for $k \leftarrow 1 \ldots N$ **do**
          Input the feature S_k of $\boldsymbol{x}$ into the IPCA I_{ik} to obtain the partial rotation R_i.
        end for
        Construct the rotation $\boldsymbol{R}$ from $< R_1, \ldots, R_N >$.
        Feed the mapped sample $(R\boldsymbol{x}, y)$ into the VFDT F_i.
      end for
    end for
  end if
  if Predict on data $\boldsymbol{x}$ **then**
    for $i \leftarrow 1 \ldots L$ **do**
      Input the mapped sample $(R\boldsymbol{x}, y)$ into the Hoeffding tree F_i to obtain the prediction $\hat{y}_i$.
    end for
    Integrate $< \hat{x}_1, \ldots, \hat{x}_L >$ to get the prediction $\hat{y}$.
  end if
end while

other methods. IPCA does not need to calculate all of the eigenvalues to find new principal components. Therefore, if the number of variables q after the transformation is much smaller than the original number of variables d, IPCA can be updated in a small amount of computational time.

Second, IPCA can easily be applied to non-stationary data streams. We can simply add a forgetting factor $0 < \lambda < 1$ to the statistics update step so that the effect of a sample obtained in the past is weakened. Because we assume that our task takes a non-stationary data stream as input, any statistics based on previously obtained samples must be forgotten when concept drift occurs.

D. Rotation Forest

RotF [17] is an ensemble method for training L decision trees independently, similar to bagging. A unique feature of RotF is that it applies a "sparse rotation" to the samples before feeding them to the base learners. This rotation is determined for each base learner using multiple PCA.

The process is as follows. Let $\boldsymbol{x} = [x_1, \ldots, x_n]$ be a sample described by n features. First, the m features are divided into N subsets. Next, bootstrap sampling is performed for each subset, and PCA is executed. The resulting N rotation matrices are integrated, creating a sparse rotation matrix $\boldsymbol{R}$. Each of the base learners in the RotF maps samples using rotation matrix $\boldsymbol{R}$ and performs training and prediction.

Bagnall et al.[1] demonstrated that RotF is an especially powerful tool when all the features are continuous. This is because the feature extraction of RotF increases the diversity of the base learners while maintaining high prediction accuracy [12].

IV. Proposed Approach

Our proposed adaptive rotation forest (ARotF) ensemble approach is an online version of RotF. It utilizes an online bagging unit, L Hoeffding Trees units, and $L \times N$ online PCA units.

Prior to sequential training, feature partitioning is executed as described in section III-D, and online PCA units are initialized. When a sample $(\boldsymbol{x}, y)$ is received, the online PCA units use it to perform incremental training. Sample $\boldsymbol{Rx}$ mapped using the updated rotation matrix $\boldsymbol{R}$ is fed to the Hoeffding tree. Prediction is performed by mapping the sample features $\boldsymbol{x}$ using the current $\boldsymbol{R}$. The Hoeffding tree units make a prediction on the basis of the mapped sample $\boldsymbol{Rx}$, and online bagging integrates the prediction results.

An obvious problem with this approach is that the incremental learning of rotation matrix $\boldsymbol{R}$ is itself a source of

concept drift. Online bagging and the Hoeffding tree are not capable of dealing with concept drift, so another method must be used to deal with the concept drift generated by a change in the rotation matrix.

The method we developed for dealing with such concept drift is ADWIN adaptive rotation forest (AARotF), which uses ADWIN bagging instead of online bagging. The base learners in the AARotF ensemble are the same as in ARotF. However, the performance of each base learner is monitored by its corresponding ADWIN. During the sequential learning process, if ADWIN detects a deterioration in the performance of a base learner, ADWIN Bagging considers that base learner to be obsolete and resets it.

Algorithms 1 shows pseudocode for the algorithm. The lines marked with * are executed only in AARotF.

V. Evaluation

We evaluated the proposed approach experimentally. We input various data streams and performed training and prediction with online bagging (bag), ADWIN bagging (abag), ARotF (arotf) and AARotF (aarotf). We fed samples from the data stream into the model one by one, with the class label hidden. First, the model predicted $\hat{y}$ on the basis of the sample. Then we input the class label for the sample and let the training run. The model was thus always tested on samples that it had not seen yet.

We compared the performance of online bagging and its extension, ARotF, and ADWIN bagging and its extension, AARotF, on each of the nine datasets. Hence, there are 18 objects for comparison. The metrics to be compared were overall prediction loss, memory consumption, total computation time, and prediction time. We used 50 Hoeffding trees as base learners, with their hyperparameters set to the same values. For all datasets, forgetting factor λ (described in section III-C) was set to 10^{-5}. This may not be the optimal value for every data stream, but it ensured that the experiment could be run multiple times and that the tuning was fair. For the methods other than ours, i.e. very fast decision tree, Online Bagging, ADWIN Bagging, as well as the experimental enviroment, we used the implementation of scikit-multiflow [13].

A. Prediction Loss

Fig. 2 shows the results for the three metrics for each method. The introduction of rotation improved the prediction accuracy for 13 of the 18 combinations of method and data stream. There were substantial improvements with the introduction of rotation in some cases. In particular, for the 4CRE-V1 dataset, ARotF and AARotF maintained very high prediction accuracy, whereas online bagging and ADWIN bagging are struggling to train, as shown in Fig. 1.

The 4CRE-V1 dataset is composed of samples generated from four Gaussian distributions on a two-dimensional feature space. The class label for each sample represents the Gaussian distribution that generated it. These Gaussian distributions change over time, which leads to severe concept drift.

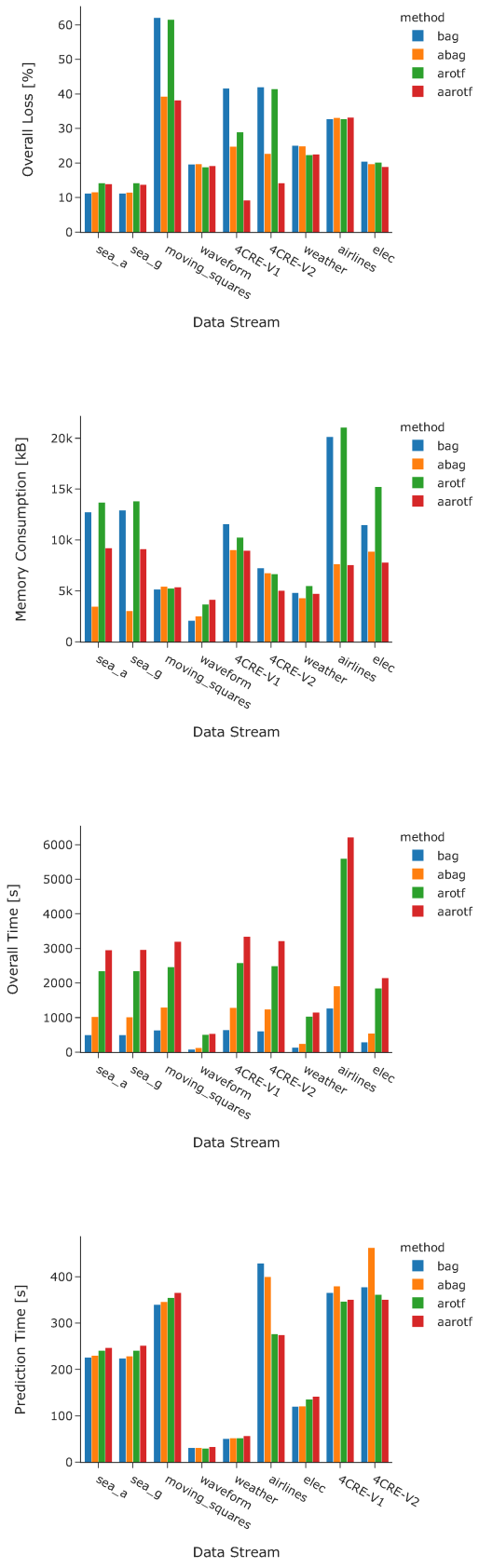

Fig. 2. Performance of each method for each data streams.

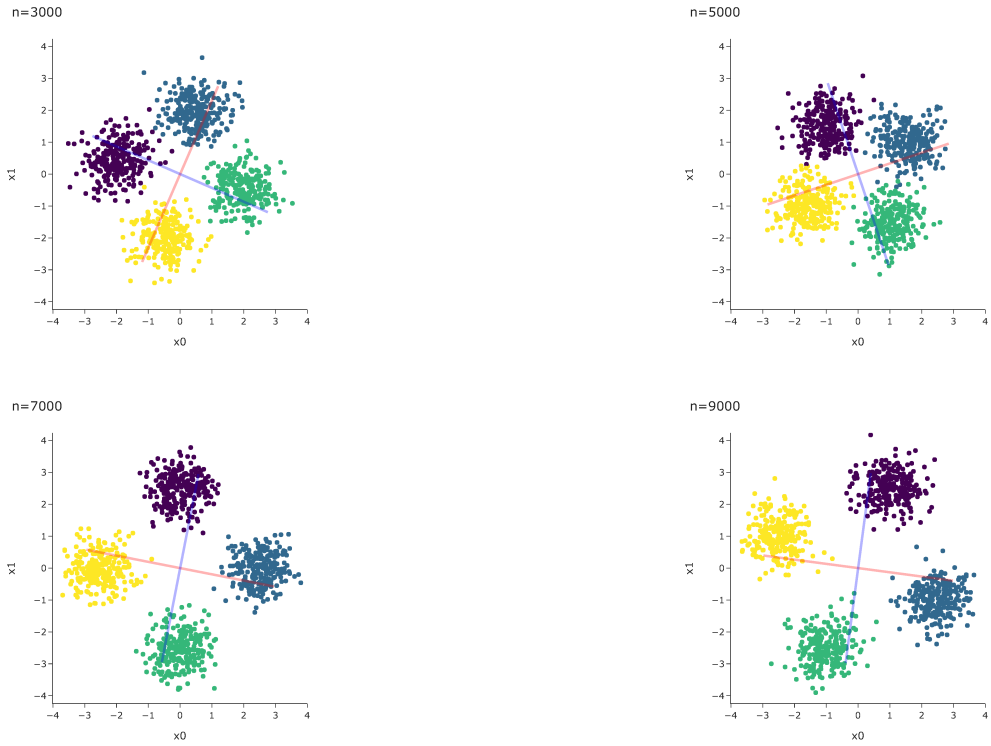

Fig. 3. Visualization of 4CRE-V2 dataset and new axes created by rotation R of single base learner. Samples are colored in accordance with class to which they belong. The distribution of samples changed over time, and rotation R followed the change well.

The distribution of the last 1000 samples and the axes rotated by $\boldsymbol{R}$ on a single tree at one moment were visualized to investigate how RotF performed for 4CRE-V1. As shown in Fig. 3, the distribution of sample changed gradually, and the IPCA instances were able to respond well to these changes. Concept drift in the 4CRE-V1 dataset is generated by gradual rotation, and the IPCA instances update the rotation matrix to follow the rotation. This enables a base learner to treat the data stream as if it were stationary. As a result, each base learner greatly improves its prediction accuracy. Since the accuracy of the entire ensemble is highly dependent on the prediction ability of each base learner, the prediction accuracy for RotF is improved.

There were some datasets for which the prediction accuracy did not improve or worsened. We attribute this to PCA not being effective in these cases. One such dataset is sea_a in which each sample has three features, and the class label is determined by whether the sum of two features exceeds a certain threshold. The process of determining the class labels is an important point. In this dataset, the samples are drawn uniformly, and then the class labels are determined on the basis of the true decision boundary. Since PCA is an unsupervised learning method, it does not work well for such datasets. In fact, attempting to perform PCA for a uniform distribution can lead to concept drift that was not originally present and not beneficial.

B. Computation Cost

The introduction of rotation improved memory consumption for 7 of the 18 combinations. In general, memory consumption was increased due to IPCA instance usage and rotation matrices. However, if an effective rotation matrix is obtained, the tree structure can be simplified, thereby reducing memory consumption. The same is true for the computation time. It increased for all datasets, but if we look at only the time taken to make a prediction, we find that it was reduced for seven combinations. This is another benefit of the simplified tree structure.

However, the time taken by learning was increased. We have concluded that the benefits of simplifying the tree structure did improve memory consumption in some of data stream, but not outweigh the additional computation time of training the IPCA units and building the rotation matrix.

VI. Conclusion

We proposed two ensemble approach for online machine learning. These methods are based on rotation forest, which is known to provide high prediction accuracy in offline learning. Our experiments demonstrated that introducing sparse rotation matrix $\boldsymbol{R}$ improves prediction accuracy compared with that of existing online ensemble approaches for the majority of data streams tested. It was particularly effective for non-stationary data streams with severe concept drift, which online bagging and ADWIN bagging struggled to handle. The amount of memory consumption and prediction time for the two methods were smaller than those for existing methods for datasets where PCA is effective. This demonstrates that sparse rotation is also useful in an online environment, which means that the proposed approach is promising for online machine learning.

One obvious problem with the proposed approach is the increase in training time. The use of online PCA methods that work faster than IPCA would reduce the time, but this would degrade the quality of rotations.

Other ways to improve the two methods are to use supervised learning instead of online PCA for the construction of the rotation matrix or to use a rotation method that takes into account the prediction accuracy of each base learner. An naïve extension based on the former idea might be supervised PCA [2]. However, it is not clear whether it would work well in an online environment. The latter solution would require updating the rotation matrix in a reinforcement-learning-like manner that would maximize the prediction accuracy of the base learner. However, reinforcement learning methods are generally computationally intensive and may not be suitable in the context of data stream mining, where the aim is real-time processing.

ACKNOWLEDGMENT

This work was partially supported by the Global Station for Big Data and Cybersecurity, a project of the Global Institution for Collaborative Research and Education at Hokkaido University.

REFERENCES

[1] Bagnall, A.J., Bostrom, A., Cawley, G.C., Flynn, M., Large, J., Lines, J.: Is rotation forest the best classifier for problems with continuous features? CoRR (2018)

[2] Barshan, E., Ghodsi, A., Azimifar, Z., Jahromi, M.Z.: Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. Pattern Recognit. **44**(7), 1357–1371 (2011)

[3] Bifet, A., Gavald, R., Holmes, G., Pfahringer, B.: Machine Learning for Data Streams: With Practical Examples in MOA. The MIT Press (2018)

[4] Bifet, A., Gavaldà, R.: Learning from time-changing data with adaptive windowing. In: Proceedings of the Seventh SIAM International Conference on Data Mining. pp. 443–448. SIAM (2007)

[5] Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., Gavaldà, R.: New ensemble methods for evolving data streams. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 139–148 (2009)

[6] Cardot, H., Degras, D.: Online principal component analysis in high dimension: Which algorithm to choose? CoRR (2015)

[7] Domingos, P.M., Hulten, G.: Mining high-speed data streams. In: Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 71–80 (2000)

[8] Dumbill, E.: A revolution that will transform how we live, work, and think: An interview with the authors of *Big Data*. Big Data **1**(2), 73–77 (2013)

[9] Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition. Springer (2009)

[10] Heusinger, M., Schleif, F.: Random projection in the presence of concept drift in supervised environments. In: Artificial Intelligence and Soft Computing - 19th International Conference, ICAISC. pp. 514–524 (2020)

[11] Johnson, W., Lindenstrauss, J.: Extensions of lipschitz maps into a hilbert space. Contemporary Mathematics **26**, 189–206 (01 1984)

[12] Kuncheva, L.I., Rodríguez, J.J.: An experimental study on rotation forest ensembles. In: Multiple Classifier Systems, 7th International Workshop, MCS. pp. 459–468 (2007)

[13] Montiel, J., Read, J., Bifet, A., Abdessalem, T.: Scikit-multiflow: A multi-output streaming framework. JMLR **19**(72), 1–5 (2018)

[14] Oza, N.C., Russell, S.J.: Online bagging and boosting. In: Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics, AISTATS (2001)

[15] Pham, X.C., Dang, M.T., Sang, D.V., Hoang, S., Nguyen, T.T., Liew, A.W.: Learning from data stream based on random projection and hoeffding tree classifier. In: 2017 International Conference on Digital Image Computing: Techniques and Applications, DICTA. pp. 1–8 (2017)

[16] Ramírez-Gallego, S., Krawczyk, B., García, S., Wozniak, M., Herrera, F.: A survey on data preprocessing for data stream mining: Current status and future directions. Neurocomputing **239**, 39–57 (2017)

[17] Rodríguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: A new classifier ensemble method. IEEE Trans. Pattern Anal. Mach. Intell. **28**(10), 1619–1630 (2006)

[18] Ross, D.A., Lim, J., Lin, R., Yang, M.: Incremental learning for robust visual tracking. Int. J. Comput. Vis. **77**(1-3), 125–141 (2008)