Fuzzy Time Windowing for Gradual Concept Drift Adaptation

Anjin Liu, Guangquan Zhang, Jie Lu
Center for Artificial Intelligence, Faculty of Engineering and Information Technology,
University of Technology Sydney,
Sydney, Australia
Anjin.liu@student.uts.edu.au, {Guangquan.zhang, Jie.lu} @uts.edu.au

***Abstract*—The aim of machine learning is to find hidden insights into historical data, and then apply them to forecast the future data or trends. Machine learning algorithms optimize learning models for lowest error rate based on the assumption that the historical data and the data to be predicted conform to the same knowledge pattern (data distribution). However, if the historical data is not enough, or the knowledge pattern keeps changing (data uncertainty), this assumption will become invalid. In data stream mining, this phenomenon of knowledge pattern changing is called concept drift. To address this issue, we propose a novel fuzzy windowing concept drift adaptation (FW-DA) method. Compared to conventional windowing-based drift adaptation algorithms, FW-DA achieves higher accuracy by allowing the sliding windows to keep an overlapping period so that the data instances belonging to different concepts can be determined more precisely. In addition, FW-DA statistically guarantees that the upcoming data conforms to the inferred knowledge pattern with a certain confidence level. To evaluate FW-DA, four experiments were conducted using both synthetic and real-world data sets. The experiment results show that FW-DA outperforms the other windowing-based methods including state-of-the-art drift adaptation methods.**

Keywords—concept drift; fuzzy concept drift adaptation; fuzzy time windowing; machine learning, adaptive learning

I. INTRODUCTION

Since the rapid development of online news and social networking services, information transmission is becoming faster and more convenient. The release of new products or new technology attracts worldwide attention, and these changes could have profound impacts on the global economic environment. A decision-maker should be sensitive to such changes and make appropriate adjustments to existing strategies. As an important decision-making tool, machine learning systems must also be able to detect and adapt to changes in learning environment. Otherwise, when a change happens, the systems may give inaccurate or misleading suggestions to users, which will result in an increasing number of decision errors.

The issue of concept drift in the machine learning field refers to the change of data distribution. Let us denote the feature vector of a data instance as X and its class label as y, and a data stream is an infinite sequence of (X, y). If the learning environment is changing, this implies that the distribution of $P(X, y)$ is changing. According to Bayes' theorem, the sources of concept drift are: i) $P(X)$ evolves with time t which can be written as $P(X|t)$; ii) a change in the conditional probability of feature X, namely $P(y|X)$. The data distribution before a learning environment change is called the old concept, and the data distribution thereafter is called the new concept.

In supervised online learning, monitoring the performance of the learning model is a well-studied approach to handle concept drift [1]. A significant drop in performance measurements, such as accuracy, will be considered as the start of a new concept. A new learning model hereafter will be trained. In this approach, the performance drop detectors, which are also called drift detectors, are critical to the overall performance of learning systems and are independent of the learning algorithms [2-4]. Because it is considered difficult to detect online distributional changes directly, most existing methods only monitor changes in particular test statistics, such as the mean or median [5]. Thus, concept drift detection is transformed to estimating whether there is any significant change in the test statistics calculated from the sequence of the performance measurements.

In practice, test statistics can only be calculated from a finite sample set. However, data streams are open-ended. Developing a method to intercept data batches from data streams is necessary. At present, time windows are the most common way to intercept data batches from data streams. Different time windows contain the data collected within different time periods. With this strategy, the drift detection problem is converted to the differential detection of the test statistics retrieved from two time-windows.

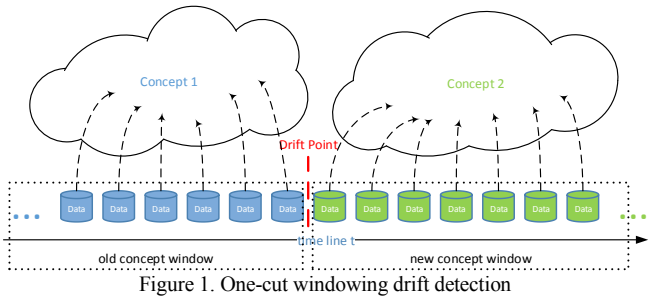

Figure 1. One-cut windowing drift detection

Although time windowing-based concept drift adaptation algorithms have delivered promising performance in many real-world applications, they have certain drawbacks. Because

the time windows are created strictly based on time steps, which assumes that concept drift occurs only at an exact time step, the boundary between windows is overly stiff, as shown in Fig.1. In fact, however, in real-world scenarios, concept drifts may last for a short period (Fig. 2), or gradually drift from one concept to another (Fig. 3). Obviously, such a one-cut windowing method will lose critical information that is shared by old and new concepts.

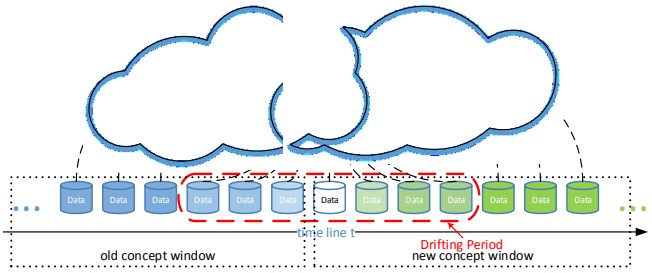

Figure 2. One-cut windowing vs. incremental concept drift

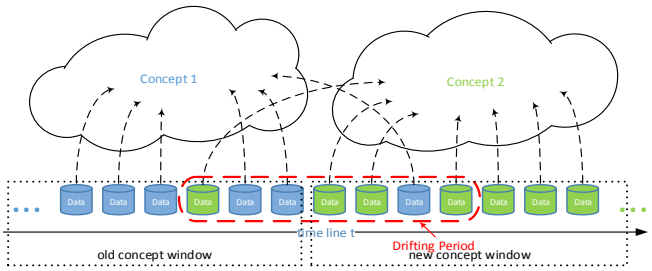

Figure 3. One-cut windowing vs. gradual concept drift

To address this problem, in this paper, we propose a novel fuzzy windowing method which allows each data instance to have a degree of membership to both old and new concepts. The usage of the shared information will be maximized, therefore achieving better performance in online machine learning. The major contributions of this paper are:

- We provide a detailed insight into windowing-based concept drift adaptation methods.
- We develop a novel fuzzy windowing method and adapt it to solve concept drift problems
- One synthetic and three real-world experiments are conducted to evaluate FW-DA, the results showing that FW-DA outperforms state-of-the-art drift adaptation methods.

The rest of this paper is organized as follows. Section II formally defines the term *concept drift*, and analyses the pros and cons of current windowing-based drift adaptation methods. Section III explains the proposed FW-DA algorithm. Section IV evaluates the proposed FW-DA. Lastly, section V concludes this paper and gives some insight into future research related to FW-DA.

II. Literature Review

A. Concept Drift

In supervised machine learning, the objective is to predict a target variable $y \in \Re^1$ in regression tasks, or the label y in classification tasks, given a set of input features X. Each data instance is a pair of (X, y). Machine learning algorithms can learn from historical data $\mathbf{D}_{train}$ and make predictions on target data $\mathbf{D}_{test}$. For static data, the learning process requires no retraining or updating, while for streaming data, it requires a verification of the consistency between historical data and new coming target data. At every time step t, after the true label is received, target data $\mathbf{D}_{test}^{t}$ will become part of the historical data, $\mathbf{D}_{train}^{t+1} = \mathbf{D}_{train}^{t} \cup \mathbf{D}_{test}^{t}$. If $\mathbf{D}_{train}^{t}$ and $\mathbf{D}_{test}^{t}$ have conflicts, then $\mathbf{D}_{train}^{t+1}$ will be inconsistent, and therefore, data cleansing for $\mathbf{D}_{train}^{t+1}$ is required. This problem is called concept drift.

Most concept drift detection algorithms are implicitly related to sudden drift detection, which assumes that a drift happens at an exact time step [6]. These algorithms do not take the drifting period (shown in Fig. 2, 3) into consideration, and this will result in bias in drift instance selection as well as the learner updating process.

B. Windowing-Based Concept Drift Adaptation

Time windowing is the most popular method to handle queries in open-ended data streams [7]. Instead of aggregating test statistics over the entire data stream, computing the test statistics from an intercepted subset in a specific time interval is more practical. Frias-Blanco [1] mentioned three relevant time windowing models: landmark, sliding and tilted windows. Landmark windows store and aggregate every instance observed in a data stream since the start point, for example, the start point is $t - i$ in Fig. 4(a). Then, successive windows share some initial points and are growing in size. Most of the time, however, we are not interested in the statistics of all past data but only that from the recent past. The simplest approach is sliding windows of fixed size, Fig. 4(b). This windowing model has a first-in-first-out data structure. Whenever a new element is observed, the oldest element in the window is removed. In tilted windows, the time scale is compressed, as shown in Fig. 4(c). The most recent data are stored inside the window with the finest detail while older information is stored at a coarser level. One example of tilted windowing is the natural tilted time window, which stores last year's data in a coarser level while storing last month's data in a fine detailed level. However, no attempt has been made to introduce fuzzy windows for concept drift detection, as shown in Fig. 4(d).

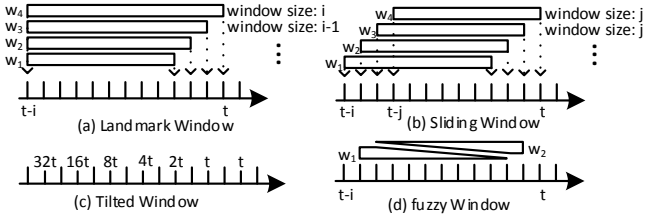

Figure 4. Time windowing models

To illustrate how the drift detection method is associated with the time windowing method, we detailed explain one of the most referenced concept drift adaptation methods, called the Drift Detection Method (DDM) [2]. In DDM, the authors use the mean and variance of the learner's accuracy as the test statistics. For each data instance, they call a prediction True if the learner outputs $\hat{y}$ equals the true label y, otherwise the prediction is called False. From this point of view, the predictions can then be seen as a random variable from

Bernoulli trials $B(n,p)$, where p is the accuracy, and n is the number of predicted data instances. Consequently, the standard deviation can be presented as $\sigma = \sqrt{p(1-p)/n}$. Considering that the probability distribution is unchanged when the context is static, then the $1-\alpha/2$ confidence interval for p with $n > 30$ examples is approximately $p \pm \alpha * \sigma$, where α is the confidence level. They defined the Warning State as $p_t + \sigma_t \geq p_{t-1} + 2 * \sigma_{t-1}$ and defined the Drift State as $p_t + \sigma_t \geq p_{t-1} + 3 * \sigma_{t-1}$, where t corresponds to the current time step. The data instances between the Warning time step and Drift time step will be used for updating the learner or training a new learner. For example, in Fig 4(a), if a Warning is detected between W_2 and W_1, and a Drift is detected between W_3 and W_4, then the data instances belong to the complement of W_2 with respect to W_4 and will be used for learner updating.

Drift adaptation algorithms that have the same windowing method are the EDDM [3], ADWIN [8], ECDD [9] and HDDM family [1]. In summary, these algorithms use the landmark windowing method to maintain a subset of data instances $\mathbf{W}_t$. Then, the authors proposed their own test statistics $s(\mathbf{W}_t)$. According to the statistical property of the proposed test statistics, they defined a Warning state and a Drift state to inform the system when and how to update the learner. Commonly, the threshold of the significance level is controlled by a parameter α. If the probability of observing a difference ϵ between two test statistics s_{t-1} and s_t is less than α (commonly $\alpha = 0.05$ for Warning state and $\alpha = 0.01$ for Drift state), the system will be notified, denoted as $P(|s_1 - s_2| \geq \epsilon) \leq \alpha$, where ϵ is the observed difference.

III. Fuzzy Windowing for Gradual Concept Drift Adaptation

In a data stream, the underlying distribution of data may change continuously over time. It is not possible to pinpoint an exact time step as the boundary between old and new concepts. To address this problem, we propose to adapt fuzzy set theory to represent the old and new concepts. Fuzzy set theory permits the gradual assessment of the membership of data instances in a concept. In addition, data instances that contribute to neither the correct classification of the old concept nor the correct classification of the new concept will be identified as a noise concept. In other words, if a data instance results in an accuracy drop of both old and new concepts, it will be considered as a noise instance.

Under this assumption, at a specific time step in a data stream, we can define three fuzzy sets: old concept, new concept, and noise concept to describe the relationships between the time-dependent distributions. Each instance in a data stream has three membership grades corresponding to these fuzzy sets. The formal definition is given below.

Definition 1. Given a data stream, at a specific time step t, $concept_t^{old} = (old_t, \mu_t^{old})$, $concept_t^{new} = (new_t, \mu_t^{new})$ and $concept_t^{noise} = (noise_t, \mu_t^{noise})$ are three fuzzy sets to describe the relationships between the data distributions before and after a learning environment drift, where μ_t^{old}, μ_t^{new} and μ_t^{noise} are the membership functions of $concept_t^{old}$, $concept_t^{new}$ and $concept_t^{noise}$.

In regard to the windowing strategy, therefore, three time-windows need be maintained accordingly, W_t^{old}, W_t^{new} and W_t^{noise}. Whenever there is a significant difference between the test statistics of W_t^{old} and W_t^{new}, we say there is a concept drift. A new learner hereafter will be trained based on W_t^{new}. Otherwise, if no drift occurs, the W_t^{old} and W_t^{new} will be updated incrementally. Thus, the proposed windowing strategy can replace the conventional crisp windowing-based drift detection algorithm. Fuzzy windowing drift adaptation consists of two parts. The first part is fuzzy windowing drift detection (FW-DD), shown in algorithm 1. The second one is fuzzy windowing drift adaptation (FW-DA), shown in algorithm 2. To simplify the drift detection process, we assume that the distribution of noise is stable, and therefore, the noise concept will not be considered for drift detection.

Algorithm 1: Fuzzy Windowing Drift Detection (FW-DD)

Input:
Data instance arrived at each time step $d_0, \cdots, d_t$
Membership function, (default: $\mu^{\mathrm{Trap}_{0.8}}$)
Test statistic, $s(\mathbf{D})$ (default: DDM $statistics$)
Statistical significance level, $P(|s_1 - s_2| \geq \epsilon) \leq \alpha$
Confidence level for warning α_w, (default: 0.05)
Confidence level for drift α_d, (default: 0.01)

Output:
State $\in$ {Stable, Warning, Drift}

1. Denote current time step as t
2. **for** $d_i : d_0, \cdots, d_t$
3. Computing the degree of membership $d_i^{W_{old}} = \mu_t^{W_{old}}(d_i)$, $d_i^{W_{new}} = \mu_t^{W_{new}}(d_i)$,where $\mu_t^{W_{old}}, \mu_t^{W_{new}}$ are the membership functions of old/new concept windows at time t
4. **end for**
5. assign membership grades as instances' weight $\mathbf{W}_t^{old} = \cup_{i=0}^{t} d_i \cdot setWeight(d_i^{W_{old}})$, $\mathbf{W}_t^{new} = \cup_{i=0}^{t} d_i \cdot setWeight(d_i^{W_{new}})$, where $\mathbf{W}_t^{old}, \mathbf{W}_t^{new}$ are corresponding weighted data instances
6. Compute current window^{old} statistics $S_t^{old} = s(\mathbf{W}_t^{old})$
7. Compute current window^{new} statistics $S_t^{new} = s(\mathbf{W}_t^{new})$
8. **if** $P(|S_t^{old} - S_t^{new}| \geq \epsilon) < \alpha_d$ **then return** Drift
9. **else if** $P(|S_t^{old} - S_t^{new}| \geq \epsilon) < \alpha_w$ **then return** Warning
10. **else** continue
11. **end if**

Instead of considering that all the data instances are equally weighted, FW-DD computes the test statistics from the data instances weighted by windows-related membership grades. The instances closer to the current time step will have higher membership grades to the new concept while the instances further from the current time step will have higher membership grades to the old concept, and vice versa. The details of DDM's test statistics and its corresponding significance level is referred in Section II. B.

In this paper, we adopt two membership functions $\mu^{\mathrm{Trap}_\lambda}$ and $\mu^{\log_\lambda}$ to describe the membership grades. The parameter λ determines the shape of the membership functions.

Definition 2. The trapezoidal membership function $\mu_t^{\mathrm{Trap}_\lambda}: concept_t^{new}$ is defined as:

$$\begin{cases} \frac{i}{\lambda \cdot |\mathbf{W_t}|} & \text{if } i \leq \lambda \cdot |\mathbf{W_t}| \\ 1 & \text{if } i > \lambda \cdot |\mathbf{W_t}| \end{cases} \quad (1)$$

where t is current time step, $|\mathbf{W}_t|$ is the size of current time window, i is the arrived time step of a given data instance.

Definition 3. The logarithm membership function $\mu_t^{log_\lambda}: concept_t^{new}$ is defined as:

$$\begin{cases} \ln\left(\frac{i(e-1)}{\lambda \cdot |\mathbf{W}_t|} + 1\right) & \text{if } i \le \lambda \cdot |\mathbf{W}_t| \\ 1 & \text{if } i > \lambda \cdot |\mathbf{W}_t| \end{cases} \quad (2)$$

where t is current time step, $|\mathbf{W}_t|$ is the size of current time window, i is the arrived time step of a given data instance.

Algorithm 2: Fuzzy Windowing Drift Adaptation (FW-DA)

Input:

Data instance arrived at each time step $d_0, \cdots, d_t$

Membership function, (default: $\mu^{Trap_{0.8}}$)

Output:

Adapted learner

1. **while** stream not end, denote current time step as t
2. **classify** d_t with current learner
3. state $=$ FW-DD$(d_0, \cdots, d_t)$
4. **if** state $=$ Warning **then**
5. warnT $= t$, default warnT $= 0$
6. **else if** state $=$ Drift **then**
7. **for** $d_i : d_{\text{warnT}}, \cdots, d_t$
8. Computing the degree of membership $d_i^{w_{new}} = \mu_{\text{warnT}}^{W_{new}}(d_i)$, where $\mu_{\text{warnT}}^{W_{new}}$ is the membership functions of new concept windows at time warnT
9. **end for**
10. Assign membership grades as instance weight $\mathbf{D}_t^{new} = \cup_{i=\text{warnT}}^{t} d_i \cdot setWeight(d_i^{w_{new}})$
11. create new learner with $\mathbf{D}_t^{new}$ and replace current leaner
12. reset time step at warnT as t_0
13. **end if**
14. **end while**

The major difference between conventional drift adaptation methods and FW-DA is that the data instances selected by FW-DA are weighted by membership grades. This process provides a more reasonable data division method to constitute old and new concepts. With regard to the membership functions, they are not limited to $\mu^{\text{Trap}_\lambda}$ or $\mu^{\log_\lambda}$, and it would be a worthy study to further investigate what kind of membership fuctions suits which drift detection methods the best.

IV. Experiments and Evaluations

To evaluate the proposed fuzzy windowing method, we applied the FW-DA on DDM [2] and ECDD [9], and tested them on one synthetic concept drift data stream and three real-world evolving data streams. The compared algorithms are DDM [2], ECDD [9], HDDM-A, HDDM-W [1] which also compute test statistics against learner accuracy. In addition, we also include one ensemble-based drift adaptation method, Weighted Majority, and an online learning drift adaptation method, Leveraging bag, for the evaluation. All the algorithms were implemented on the MOA platform [10]. The experiments were conducted on a cluster node with 3.4GHz 8 cores CPU and 32GB RAM. For all experiments, the membership function parameter λ was set to 0.8; and the parameters of the compared algorithm were set as the default values as suggested by their authors. The evaluation metrics are the average accuracy, precision, recall, f-score and running time.

As the test statistics of these algorithms are all based on the accuracy of the classification results, to avoid the influence of multiclass classification errors, we only evaluate our methods on binary classification problems.

A. Experiment 1: SEA Concept Drift Data Streams

To evaluate the performance of FW-DA, the first experiment was conducted on the most popular synthetic concept drift streams. This was proposed in [11] and has been widely used in concept drift research.

SEA stream is a benchmark data set for evaluating concept drift-related algorithms [12-14]. In this data set, there are four blocks of data with different concepts. Each block contains 5000 random points within the 3D feature space. The three features have values randomly generated in the range $[0, 10]$, and only the first two features are relevant to the label. In each block, a data point belongs to class1, if $x_1 + x_2 < \theta$, where x_1 and x_2 represent the first two features, and θ is a threshold value for the two classes. Threshold values for the four data blocks are 8, 9, 7 and 9.5 in sequence.

TABLE I. The performance of Different Drift Adaptation Algorithms on SEA Streams.

Algorithm	Acc.	Prec.	Rec.	F1	Time (ms)
DDM	0.8615 ± 0.0595	0.8640 ± 0.0614	0.8407 ± 0.0721	0.8522 ± 0.0656	**1004.67 ± 57.36**
FW-DDM-Trap	**0.8685 ± 0.0118**	**0.8706 ± 0.0123**	**0.8492 ± 0.0159**	**0.8598 ± 0.0119**	1005.10 ± 66.19
FW-DDM-Log	0.8671 ± 0.0130	0.8691 ± 0.0127	0.8477 ± 0.0184	0.8583 ± 0.0130	1238.50 ± 2338.81
ECDD	0.8486 ± 0.0182	0.8464 ± 0.0193	0.8301 ± 0.0223	0.8382 ± 0.0200	1005.07 ± 60.18
FW- ECDD-Trap	0.8477 ± 0.0192	0.8451 ± 0.0202	0.8295 ± 0.0226	0.8372 ± 0.0207	1005.03 ± 61.05
FW- ECDD-Log	0.8478 ± 0.0189	0.8452 ± 0.0199	0.8295 ± 0.0222	0.8373 ± 0.0202	1005.17 ± 58.16
HDDM-A-Test	0.8627 ± 0.0413	0.8644 ± 0.0431	0.8428 ± 0.0461	0.8535 ± 0.0436	1005.43 ± 60.79
HDDM-W-Test	0.8484 ± 0.0473	0.8494 ± 0.0456	0.8269 ± 0.0586	0.8380 ± 0.0500	1005.13 ± 58.18
Weighted Majority	0.8362 ± 0.0117	0.8400 ± 0.0107	0.8100 ± 0.0221	0.8247 ± 0.0149	1022.30 ± 123.82
LeveragingBag	0.8623 ± 0.0202	0.8646 ± 0.0245	0.8418 ± 0.0244	0.8530 ± 0.0225	1039.27 ± 1057.26

To comprehensively compare the performance of different drift adaptation algorithms, we generated 50 SEA concept data streams with the same configurations, and illustrate the mean of accuracy, precision, recall and f-score with the corresponding standard deviation as shown in Table I. From the results, we can see that the proposed fuzzy windowing method boosted the overall performance of both DDM and ECDD. FW-DDM-Trap even outperformed HDDM-A-Test to become the best performing algorithm.

B. Experiment 2. Electricity Data Set

In this experiment, we compare our methods with other algorithms using a real-world data set, Electricity Data. Electricity Data contains 45,312 instances, collected every thirty minutes from the Australian New South Wales Electricity Market between 7 May 1996 and 5 Dec 1998. In this market, prices are not fixed and are affected by demand and supply in the market. This data set contains 8 features and 2 classes and has been widely used for concept drift adaptation evaluation. The classification results are shown in Table II.

TABLE II. THE PERFORMANCE OF DIFFERENT DRIFT ADAPTATION ALGORITHMS ON ELEC DATA SET

Algorithm	Acc.	Prec.	Rec.	F1	Time (ms)
DDM	0.8118	0.8094	0.8031	0.8062	**1002**
FW-DDM-Trap	0.8593	0.8560	0.8560	0.8560	1004
FW-DDM-Log	0.8431	0.8394	0.8397	0.8395	2005
ECDD	0.8676	0.8643	0.8650	0.8646	1005
FW- ECDD-Trap	**0.8691**	**0.8657**	**0.8669**	**0.8663**	2067
FW- ECDD-Log	0.8685	0.8650	0.8665	0.8658	2136
HDDM-A-Test	0.8492	0.8462	0.8446	0.8454	3063
HDDM-W-Test	0.8409	0.8374	0.8367	0.8371	1004
Weighted Majority	0.7336	0.7571	0.7018	0.7284	2151
LeveragingBag	0.7888	0.7922	0.7725	0.7822	2006

As shown in Table II, both μ^{Trap_λ} and μ^{Log_λ} are beneficial to conventional drift adaptation algorithms. Although the improvement of FW-ECDD-Trap and FW-ECDD-Log may not be considered significant, they have no adverse impact on the original algorithms. As for FW-DDM-Trap, it improved the overall performance significantly and surpassed HDDM-A-Test and HDDM-W-Test. Without considering the limitation of DDM on this data set, the results of fuzzy windowing methods are promising.

C. Experiment 3: Airline Data Set

The second real-world data set used is the Airline data set. It consists of flight arrival and departure details for all commercial flights within the USA, from October 1987 to April 2008. This data set was originally proposed for regression problems in the Data Expo competition in 2009. Then it was modified by the MOA team [10] for prediction analysis. Each data instance has 7 features and 2 classes $\{delay, not\ delay\}$. It has 539388 records in total.

As shown in Table III, the FW-based algorithms have a similar effect on the original work as in experiment 2. The performance of both DDM and ECDD was improved and FW-DDM algorithms achieved more improvement than FW-ECDD. FW-DDM-Trap outperformed all other algorithms in accuracy while it was only a little lower than HDDM-A-Test in f-score. In relation to ECDD-related algorithms, it seems that their test statistics are not good enough for this data set. Obviously, ECDD has its own limitations in this data set which also limited the power of fuzzy windowing.

TABLE III. THE PERFORMANCE OF DIFFERENT DRIFT ADAPTATION ALGORITHMS ON AIRLINE DATA SET

Algorithm	Acc.	Prec.	Rec.	F1	Time (ms)
DDM	0.6533	0.6526	0.6362	0.6443	7006
FW-DDM-Trap	**0.6725**	0.6715	0.6581	0.6647	16018
FW-DDM-Log	**0.6725**	**0.6718**	0.6579	0.6647	11028
ECDD	0.6364	0.6309	0.6282	0.6295	7004
FW- ECDD-Trap	0.6369	0.6314	0.6286	0.6300	10005
FW- ECDD-Log	0.6376	0.6321	0.6292	0.6306	10005
HDDM-A-Test	0.6722	0.6688	**0.6612**	**0.6650**	**6004**
HDDM-W-Test	0.6534	0.6484	0.6442	0.6463	**6004**
Weighted Majority	0.6455	0.6532	0.6216	0.6370	6034
LeveragingBag	0.6654	0.6609	0.6565	0.6587	25009

D. Experiment 4. Spam Filtering Data

In this experiment, we compare our methods with other algorithms on Spam Filtering Data. Spam Filtering Data is a collection of 9324 email messages derived from the Spam Assassin collection, which is available at http://spamassassin.apache.org/. This data set contains 39916 features, and 9324 emails (around 20% spam emails and 80% legitimate emails). It has been considered a typical gradual drift data set since the work in [15]. According to Katakis's work [15], 500 attributes were retrieved using the chi-square feature selection approach. Table IV lists the classification results of the spam data.

TABLE IV. THE PERFORMANCE OF DIFFERENT DRIFT ADAPTATION ALGORITHMS ON SPAM FILTERING DATA SET

Algorithm	Acc.	Prec.	Rec.	F1	Time (ms)
DDM	0.8954	0.8786	0.8378	0.8577	2005
FW-DDM-Trap	0.8965	0.8860	0.8329	0.8586	3005
FW-DDM-Log	0.9003	0.8836	0.8465	0.8647	4079
ECDD	0.8884	0.8709	0.8252	0.8474	2005
FW- ECDD-Trap	0.9127	0.8978	0.8673	0.8823	3009
FW- ECDD-Log	0.8964	0.8859	0.8327	0.8585	2004
HDDM-A-Test	0.9079	0.8814	0.8747	0.8781	**2003**
HDDM-W-Test	**0.9165**	**0.9015**	0.8742	0.8876	**2003**
Weighted Majority	0.9066	0.8748	0.8828	0.8788	2024
LeveragingBag	0.9173	0.8862	**0.9021**	**0.8941**	6009

The results in Table IV show that the FW-based algorithms outperform conventional windowing methods. FW-ECDD-Trap surpasses HDDM-A-Test, however, HDDM-W-Test is still the best one. This may be due to the drift adaptation algorithm itself which is beyond the scope of this study. However, it is no doubt that fuzzy windowing methods can contribute to these algorithms in real-world scenarios.

This analysis clearly shows that fuzzy windowing methods improve the corresponding original work. With these

improvements, the tested drift adaption algorithms can even surpass some algorithms that used to have superior performance. Fuzzy windowing method not only improved the original work but also outperformed one of the state-of-the-art drift adaptation algorithms. Although the running time of fuzzy windowing methods is slightly higher than the others, considering the size of the data sets, it is still acceptable.

V. Conclusion and Future Work

In this study, we first comprehensively reviewed the state-of-the-art windowing-based concept drift adaptation algorithms. Then, we highlighted the deficiencies of the current methods. Against the shortage of current methods, we introduced fuzzy set theory to describe the concept before and after an environmental drift. As per the given definitions, we proposed FW-DA to improve the current concept drift adaptation algorithms. By carrying out a systematic evaluation, we concluded that FW-DA is beneficial to both concept drift detection and adaptation and it is evident that using fuzzy sets to describe concepts in data streams is more in line with real-world scenarios.

Further work includes introducing the membership function for noise windows, and investigating the relationship between the test statistics and membership functions. These studies would help to further refine the windowing-based concept drift adaptation methods.

VI. Acknowledgements

This work is supported by the Australian Research Council (ARC) under discovery grant DP150101645. Also, the authors would like to thank the anonymous reviewers for their valuable feedback and all members of the Decision Systems and eService Intelligence laboratory of University of Technology Sydney for Discussion.

References

[1] I. Frias-Blanco, J. d. Campo-Avila, G. Ramos-Jimenes, R. Morales-Bueno, A. Ortiz-Diaz, and Y. Caballero-Mota, "Online and Non-Parametric Drift Detection Methods Based on Hoeffding's Bounds," *IEEE Transactions on Knowledge and Data Engineering,* vol. 27, pp. 810-823, 2015.

[2] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Proceedings of the Seventeenth Brazilian Symposium on Artificial Intelligence*, Sao Luis, Maranhao, Brazil, 2004, pp. 286-295.

[3] M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno, "Early drift detection method," presented at the Fourth International Workshop on Knowledge Discovery from Data Streams, 2006.

[4] A. Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," in *Proceedings of the Seventh SIAM Conference on Data Mining*, 2007, pp. 443-448.

[5] G. J. Ross, D. K. Tasoulis, and N. M. Adams, "Nonparametric monitoring of data streams for changes in location and scale," *Technometrics,* vol. 53, pp. 379-389, 2011.

[6] N. Lu, G. Zhang, and J. Lu, "Concept drift detection via competence models," *Artificial Intelligence,* vol. 209, pp. 11-28, 2014.

[7] J. Gama and M. Gaber, *Learning from Data Streams: Processing Techniques in Sensor Networks*, pp. 38-40, Springer Verlag, 2007.

[8] A. Bifet and R. Gavaldà, "Adaptive learning from evolving data streams," in *Proceedings of the Eighth International Symposium on Intelligent Data Analysis*, 2009, pp. 249-260.

[9] G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand, "Exponentially weighted moving average charts for detecting concept drift," *Pattern Recognition Letters,* vol. 33, pp. 191-198, 15 Jan 2012.

[10] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive online analysis," *The Journal of Machine Learning Research,* vol. 99, pp. 1601–1604, 2010.

[11] W. N. Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification," in *Proceedings of the Seventh ACM International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, 2001, pp. 377-382.

[12] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Transactions on Neural Networks,* vol. 22, pp. 1517-31, Oct 2011.

[13] A. Liu, G. Zhang, and J. Lu, "Concept Drift Detection Based on Anomaly Analysis," in *International Conference on Neural Information Processing*, 2014, pp. 263-270.

[14] N. Lu, J. Lu, G. Zhang, and R. L. De Mantaras, "A concept drift-tolerant case-base editing technique," *Artificial Intelligence,* vol. 230, pp. 108-133, 2016.

[15] I. Katakis, G. Tsoumakas, E. Banos, N. Bassiliades, and I. Vlahavas, "An adaptive personalized news dissemination system," *Journal of Intelligent Information Systems,* vol. 32, pp. 191-212, 2009.