2017 IEEE 6th International Congress on Big Data

A Hybrid Learning Framework for Imbalanced Stream Classification

Wenbin Zhang, Jianwu Wang
University of Maryland, Baltimore County, MD, USA
{wenbinzhang, jianwu}@umbc.edu

***Abstract*— The pervasive imbalanced class distribution occurring in real-world stream applications, such as surveillance, security and finance, in which data arrive continuously has sparked extensive interest in the study of imbalanced stream classification. In such applications, the evolution of unstable class concepts is always accompanied and complicated by the skewed class distribution. However, most of the existing methods focus on either class imbalance problem or non-stationary learning problem, the combined approach of addressing both issues has enjoyed relatively little research. In this paper, we propose a hybrid framework for imbalanced stream learning that consists of three components: classifier updating, resampling and cost sensitive classifier. Based on the framework, we propose a hybrid learning algorithm to combine data-level and algorithm-level methods as well as classifier retraining mechanics to tackle class imbalance in data streams. Our experiments using real-world datasets and synthetic datasets show that our proposed hybrid learning algorithm can have better effectiveness and efficiency.**

***Keywords*—Class imbalance, concept drift, data stream mining, hybrid learning.**

I. INTRODUCTION

More than two decades of continuous development learning has produced a variety of algorithms for classification. Traditional classifier identifies a suitable hypothesis to make good prediction from a stationary environment, the instances of which belong to an underlying distribution defined by a generating function. This static dataset is therefore assumed to be bounded and the algorithm can afford to read the data several times to learn the relevant concepts pertaining to the underlying generating function. However, the dynamic environment in many real-world applications, such as health care [1], intrusion detection [2] or financial businesses [3], can change the target concept over time. This poses a difficulty for traditional learning algorithm as the data is non-stationary and arrives over time in streams of instances (incremental) or batches (batch) instead of being available from the beginning. Learning under such conditions requires classifier to be trained on the information before time step *t* to predict new instances arriving at time step $t+1$, and to update incrementally by leveraging the newly available data at time step $t+1$ while simultaneously maintaining the performance of the classifier on the information before time step t. Furthermore, these scenarios also present the distribution of examples is skewed since representatives of some of classes, called *minority classes*, are represented by a negligible number of instances pertaining to other classes, called *majority classes*, that are considered. At the same time although minority classes are rare, they are usually the important case of the study as it may carry important and useful knowledge. For instance, correctly identify fraud, the number of which is severely less compared with the amount of normal customers, in the credit card fraud identification is more important, and hence we require methods to improve its recognition accuracy. This domain, known as imbalanced learning, requires to take dataset imbalance into consideration in classifier designing, otherwise the traditional classifier can be easily fooled, being overwhelmed by the majority classes and ignoring the minority classes.

The concept drift results from dynamic environment in real-world application always arises along with class imbalance and further exacerbated each other. In recent years, a number of studies have been proposed to address variation in the underlying function [4] and to tackle class imbalance [5], respectively. However, the combined research, especially in streaming data settings, has received relatively little attention [6]. To the best of our knowledge, there is no study yet combining data-level and algorithm-level methods as well as classifier retraining mechanics for the classification of imbalanced streaming data. This paper describes a novel hybrid learning framework in dealing with imbalanced stream classification. In our approach, concept drift in stream is detected considering each single-class performance along with class percentages calculated in dynamically changing environment to decide when to invoke the hybrid learner to deal with concept drift and class imbalance. With the information of concept drift and class imbalance, the hybrid learner applies sampling techniques to adjust the learning bias from majority towards the minority, and chooses to either be confident with the current learner or update the learner with the new concept learned from current stream by ensembling cost-sensitive classifiers. Minority instances from previous streams are also being kept track of simultaneously with the goal of a performance balance among classes.

In summary, the contributions of the papers are as follows:

1) A novel hybrid learning framework is proposed to address non-stationary learning problem and class imbalance issue concurrently. This customizable framework is flexible in incorporating a variety of different methods for the task at hand.

2) According to the proposed framework, a hybrid learning algorithm is designed to integrate resampling, cost sensitive as well as classifier retraining mechanics in dealing with

978-1-5386-1996-4/17 $31.00 © 2017 IEEE
DOI 10.1109/BigDataCongress.2017.70

imbalanced stream classification.

3) As the first work that formally introduces cost sensitive learning in streaming data settings, experiments are carried out on both real-world datasets and synthetic datasets to compare the efficiency performance from resampling, cost sensitive and resampling in conjunction with cost sensitive perspectives. The experimental results conclude the positive impact of combining data-level and algorithm-level methods on effectiveness and efficiency, which are two essential targets of stream learning applications.

The remainder of the paper is organized as follows. Section II and III review theoretical background regarding concept drift and class imbalance as well as related works. Section IV proposes the learning framework for imbalanced stream classification and describes it in details. The experimental study on both real world and synthetic datasets is performed and the obtained results are analyzed in Section V. Finally, Section VI states conclusions and future works.

II. Background

When learning from streaming data, concept drift involves changing the concept of a given target. Assuming at time step t a sequence of labelled data instances $\zeta = \{(\mathrm{X}_0, \mathrm{y}_0), (\mathrm{X}_1, \mathrm{y}_1), \ \ldots, \ (\mathrm{X}_t, \mathrm{y}_t)\}$ presented in chronological order, where X_i is a p-dimensional feature vector and each instance has a corresponding class label y_i. The learning algorithm is trained given ζ and predicts unlabeled instance X_{t+1} at time step $t+1$. Once y_{t+1} is predicted, the actual class label of X_{t+1} becomes available and new instance X_{t+2} at time $t+2$ arrives afterwards. Concept drift occurs when the target concept of time t is different from time $(t+1)$'s. Depending on the rate at which concept drift presents, the drift may be considered as gradual or abrupt. In gradual drift, the change between two concepts happens with a smooth transition. Abrupt drift, on the contrary, the change between concepts suddenly switch within a definite time period. As concepts change over time, if there are instances where a concept reoccurs, this is called reoccurring drift. The variation in the underlying function, from Bayes' theorem's perspective, can result from changes occur in three major ways: the prior probability of observing each class $p(y_i)$, the conditional probability that an instance drawn from class y_i would be X_i and the posterior distributions of class membership $p(y_i|X_i)$. Kelly et al. claim that it is only the change in posterior probability that is important [22]. Learning from data streams therefore requires a learning system being able to remain stable on previously learned and not outdated concepts while incrementally update the knowledge learn on new data with possible concept drift. For more details, refer to [7].

III. Related Work

In recent years, a number of methods have been proposed to learn in the presence of concept drift. These approaches fall largely in three groups: 1) adaptive-based approaches use restricted or expanded data to build the classifier that predicts new instances in some region of the feature space [8]; 2) modification methods, which are classifier agnostic, select or weight the instances and the outdated training instances can be discarded based on their weights [9]; 3) an ensemble of classifier produces outputs of several classifiers and combine them to determine a final classification [10]. Considering the advantages of above three techniques in streaming data, our proposed framework encompasses keeping track of up to date minority instances in building the learning classifier, and discarding old examples as well as trained classifiers when the most recent data stream indicates a change in the distribution. In addition, trained classifiers are combined to create more accurate classifies to overcome concept drift.

Learning from data stream also suffers from class imbalance, a large number of approaches have been proposed with respect to this specific issue. These methods fall into three main categories. First, data-level solutions [5] concentrate on modifying the original collection of training set in order to reduce or eliminate the extent of datasets imbalance, such as generating new samples for the minority class (oversampling), getting rid of objects from majority class (undersampling) and combining both methods to change the distribution balance of original data. Second, algorithm-level solutions [12] modify existing algorithms to alleviate their bias towards majority class for the sake of benefiting the classification of the minority class, such as cost-sensitive methods [13], which incorporate different misclassification penalty for each of considered class so that the classifiers pay more attention to underrepresented set of instances, and one class learning [14], which creates a data description to concentrate only on a single group of examples instead of bias towards any group. Last, hybrid solutions [15] are proposed that combine the strong points of previous two methods to address classification with uneven data representation, Wozniak et al. [16], for instance, propose notion of hybridization of data-level solutions and classifier ensembles, resulting in a robust and efficient learning algorithm. In our proposed framework, the distributions of positive and negative objects are monitored to determine when to reduce imbalance rate by data-level solutions, the imbalance rate is further reduced according to varying error costs that introduced by MetaCost [13], a procedure that make error-based classifier cost-sensitive.

The fact that unstable class concepts in numerous real-life applications also suffer from skewed class distribution has drew increasing attention which seeks to tackle both issues simultaneously. However, as noted in [6], there is relative paucity of such research into imbalanced streaming data, and the existing methods mainly focus on either of them. Lichtenwalter et al. propose a method, called Boundary Definition (BD), to define the class boundary in propagating instances misclassified by the current model [17], and the performance of ensemble members that built on such boundary shows improvement. Recently, a selectively retrain approach based on clustering updates the ensemble classifier with the base classifier that trained on the most up-to-date chunk [18]. Our proposed framework outlines a new approach that utilizes a hybrid learning framework in imbalanced stream classification setting.

IV. Hybrid learning framework

Our approach stems from the common idea, that of the moving window [9], for learning from streaming data sources. This idea, as its name implies, is about maintaining a classifier that can be updated incrementally from a moving window of newly available instances. We integrate imbalance learning when applying this idea to address both concept drift and class imbalance concurrently. Our proposed learning framework consists three main components, each component in constant dialogues with other components for the update-to-date status of data streams and takes corresponding response accordingly. Figure 1 shows the workflow of the proposed learning framework.

The first component, called Classifier Updating, is designed to trigger the classifier updating mechanism based on the performance of each single-class. Most methods proposed so far are based on overall prediction error made by the learner to detect concept drift and to take the corresponding action adaptively when its performance degrades [4]. The underlying assumption behind those methods believes the functioning deterioration of learner, that trained on out-of-date concepts, is result from the incapability of synthesizing current drifting concept. However, this assumption is not appropriate when taking imbalance characteristic into data streams consideration. Take the simplest binary classification of a dataset which consists of 99 percent of majority instances and 1 percent of minority instances as an example, a naïve classifier can get an accuracy of 99 percent by simply classifying all instances into majority category. Although the superb face value is overly outperforms random guess, it cannot reflect the performance on minority class and therefore contributes too little to discover the drift in imbalance data stream. In our framework, we evaluate prediction errors for each individual class to provide more adequate information on the functionality of trained classifier, based on this information, the updating status of ensemble classifier is further determined. Specifically, when a new batch of data stream arrives, prediction is made based on the current learner and corresponding performance metrics of both minority and majority class are computed. On the condition that recalls from both classes are better than predefined thresholds, the training data that the current learner built on still carries with up-to-date concepts, learner update is therefore considered not necessary for computational efficiency. On the other hand, when prediction error that attributes to either minority or majority class is greater than predefined threshold or both of them go below the satisfaction border line, training a new classifier for the new concept is triggered.

The second component, called Resampling, gets involved in the learning mechanism immediately after the trigger of updating classifier. The imbalance status is also monitored each time a new batch of data stream arrives. Based on this captured information, our framework determines whether to initiate resampling. Other than under-representativeness, poor performance is also results from other factors such as the complexity of data distribution [19]. Therefore, it is not necessary to complicate the learning procedure if the data is evenly distributed. In our framework, resampling techniques will be invoked to give minority class more focus only on the condition that class imbalance is recognized. In this paper, the synthetic minority oversampling technique (SMOTE) [11] is employed to increase the chance of training minority class instances, and the chance of training majority class examples is decreased by undersampling. Here, undersampling and oversampling are used in conjunction with resample heuristics to increase precision while minimally affecting recall. According to the prediction results made by the learner on the new batch of data stream, the heuristic function first divides instances into four subsets: misclassified minority instances, misclassified majority instances, correctly classified minority instances and correctly classified majority instances. The heuristic function ranks the first two subsets with higher weights with the idea of propagating misclassified instances defines class boundary better than correctly classified instances. Therefore, SMOTE is only applied on misclassified minority examples to give more focus on these instances, the chance of training majority class is decreased by undersampling correctly classified majority instances as they are less likely in helping with precision improvement. In the meanwhile, misclassified majority instances remain unchanged to avoid too much information loss, so does correctly classified minority instances. These four subsets after resampling comprised the new training set for updating classifier.

Different from existing approaches whose classifiers minimize zero-one loss in the streaming data settings [6], the third component of our framework, in order to further reduce the imbalance rate, wraps a "meta-learning" stage around the error-based classifier such that the classifier effectively minimizes cost, with a bound on the sum of zero-one loss. This procedure makes error-based classifier cost-sensitive and is known as MetaCost [13]. In this circumstance, the Bayes optimal prediction for a given example x is the class i that minimizes the conditional risk defined as $R(i|x) = \sum_j P(j|x)C(i,j)$, where $P(j|x)$ represents the probability of each class j for the given example x and $C(i,j)$ is the cost matrix being the cost of misclassifying the example with actual class j as class i. In our approach, there is no cost for correct classification of either class and the cost of misclassifications are set according to the imbalance rate monitored from the second component. Considering the learning difficulty of minority instances due to their sparsity, a queue structured minority window is maintained in the framework as the collection of minority instances from the previous streams. This design ensures first added minority examples will be the first one to be removed when size limit reaches so as to better address minority-class concept learning. After the triggering of training a new classifier, the instances from minority window along with dataset from the second component are used to train a

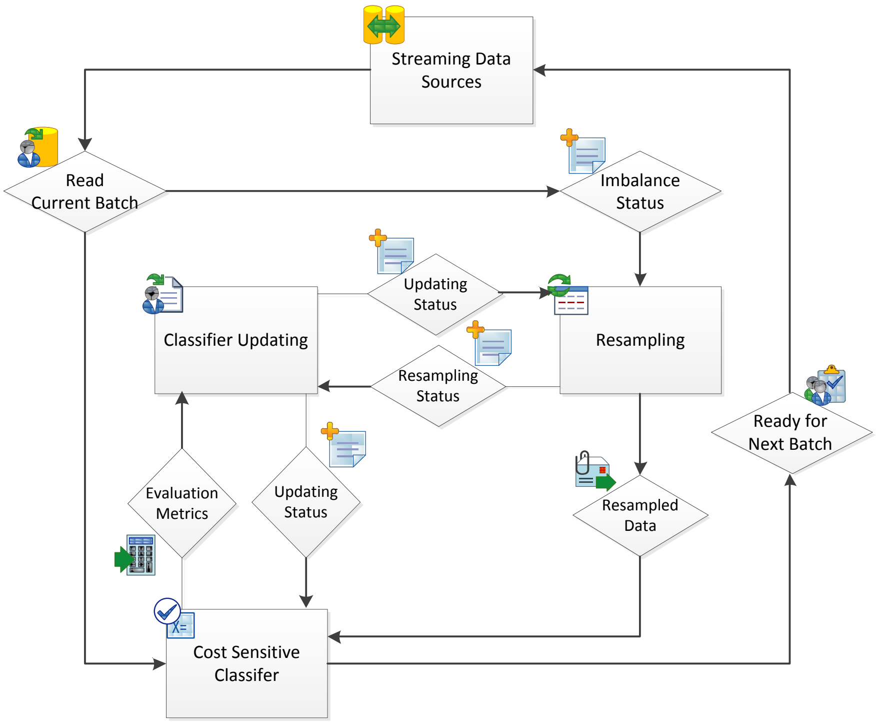

Fig. 1: A hybrid learning framework in dealing with imbalanced stream classification.

new cost-sensitive learner for the new concepts from current stream. Then, the strength of newly discovered predictive relationships on the current batch of data stream is assessed. The newly trained classifier will be added to contribute for the ensemble learner if its recalls from both classes are better than predefined thresholds, framework completes the learning of current stream with updating minority window. Otherwise, the new learned classifier is discarded and only minority window updating is performed. The ensemble learner is also designed in the queue structure.

Other than the resampling techniques that employed in this work, a variety of different methods [5] from the resampling community can be embedded into our proposed framework to customizably solve the problems according to the characteristics therein and from different perspectives. This framework is also flexible in the way that can be modified so as to adapt to different circumstances. For instance, when dealing with a task that prior experience or knowledge demonstrates the existence of imbalance feature all over data streams, the module that monitors imbalance status can be deactivated for better efficiency. Similarly, when the hypothesis from current data stream is particularly being concerned, learner update is therefore preferred upon the arrival of every batch of new data stream. Under this circumstance, classifier updating unit is expected to be turned off and its dialogues with other components can be muted, the remaining components communicate directly without the intervention of classifier updating component.

Based on the above hybrid learning framework, we propose a hybrid stream learning algorithm, which is detailed in Algorithm 1. Given the selected values of sliding window (SW) size S_s, minority window (MW) size S_m, ensembled cost-sensitive classifier window (CW) size S_c, satisfactory classifier accuracy threshold α, imbalance threshold β as well as the data stream sources (D) as input, the algorithm predicts the class label for each instance and outputs the values of evaluation metrics for each sliding window. It consists of four parts. In the first part (line 1-3), it simulates the arrival of data streams with fixed window size. The second part (line 5-14) deals with the arrival of data streams when CW is empty. Line 6-8 train an initial classifier on the current window and determine whether to add it to CW considering its prediction errors for each individual class. Line 9-14 retrain a classifier based on the resampled window instances if the current window is imbalanced and determine whether the retrained classifier should be added into CW. The third part (line 15-24) further processes the data streams when CW contains at least one trained classifier. Line 16 and 17 find whether classifier updating and resampling are necessary, respectively. Line 20-24 update the CW with the retrained classifier. The fourth part (line 25-28) works with the maintenance of MW. The algorithm stops until the data streams come to an end (In Algorithm 1, acc_{min} and acc_{maj} are abbreviations for the accuracy of minority instances and majority instances, respectively).

Algorithm 1 Hybrid Learning Algorithm

```
while D has more instances do
    if $|SW| < S_s$ then
        Add instance to SW;
    else
        if $CW == \emptyset$ then
            Train the initial cost-sensitive classifier with data from SW;
            if $acc_{min} > \alpha \&\& acc_{maj} > \alpha$ then
                Add the trained classifier to CW;
            else
                if imbalance rate of SW is less than $\beta$ then
                    Resampling SW;
                    Retrain an initial cost-sensitive classifier with resampled SW;
                    if $acc_{min} > \alpha \&\& acc_{maj} > \alpha$ then
                        Add the trained classifier to CW;
        else
            if $acc_{min} < \alpha \&\& acc_{maj} < \alpha$ then
                if imbalance rate of SW is less than $\beta$ then
                    Resampling SW;
                    Retrain a new cost-sensitive classifier with resampled SW;
                if $acc_{min} > \alpha \&\& acc_{maj} > \alpha$ then
                    if $|CW| < S_c$ then
                        Add the new trained classifier to CW;
                    else
                        Replace the oldest classifier with the new trained classifier;
        if $|MW| < S_m$ then
            Add minority instances from SW to MW;
        else
            Replace the oldest minority instances with new minority instances.
```

V. Experiments

This section presents an empirical study designed to evaluate the performance of our proposed hybrid learning algorithm on imbalanced data streams. The evaluation is based on efficiency and effectiveness measurement using two real datasets in addition to synthetic data. For each dataset, we run our proposed hybrid algorithm along with resampling based approach and cost sensitive oriented method. Boundary Definition (BD) that discussed in Section III is served as the representative of resampling based approach. To our best knowledge, we have not seen cost sensitive work in imbalanced streaming data setting, so we apply our framework without involving the resampling component as comparison, referred as CostSensitive.

A. Description of Datasets

In our experiment, we evaluate the effectiveness and efficiency of our proposed framework using two real-word datasets in addition to synthetic data which simulate three possible concept drift occurrences. We select two benchmark datasets, which are publicly available and have been used by existing works for data stream classification [7], [14], [17]. The first dataset, the electricity dataset, is described in detail by I. Žliobaitė [20]. Briefly, this data was collected from the Australian New South Wales Electricity Market; there are 45,312 instances. The task is to identify the change of the unfixed price relative to a moving average of the last 24 hours. The second dataset, the airline dataset, contains flight arrival and departure details for the commercial flights within the USA [17]; there are 539,383 instances. The class label identifies whether a given flight will be delayed, given the information of the scheduled departure. Existing works address these two datasets from streaming learning perspective and ignores imbalance issue as datasets are inherently unskewed distributed. We render them as imbalanced data streams by removing certain number of instances from designated classes arbitrarily, then randomizing the order of the remaining instances and processing them in sequence. As one of the data preprocessing steps, the numeric attributes of the electricity dataset are centralized as the computation of normalization in order to compensate for technical difference in collecting data and enable informative comparisons between different instances. Characteristics of these two processed datasets are shown in Table I.

In addition to the two real-world datasets, simulation data are also used in our experiments. Simulation imitates a real-world process without the need to carry out a pilot test while permitting a sufficient understanding of the process by synthetizing dataset with desired characteristics. We design a data generation algorithm to specifically address imbalanced streaming data. In particular, we use Random RBF (Radial Basis Function), Rotating Hyperplane, and Random RBF along with Rotating Hyperplane generators to generate three possible concept drift occurrences, gradual drift, abrupt drift and reoccurring drift, respectively [24]. The characteristics of simulation data are also available in Table I.

TABLE I: Characteristics of Experimental Datasets.

Dataset Name	No. of Instances		No. of Attributes		Imbalance	
	All	Minority	Nom	Num	Original	Now
Airline	397,531	98,412	4	3	1.25:1	4:1
Electricity	26,075	5,210	1	7	1.4:1	5:1
Gradual Drift	100,000	23,464	0	20	N/A	4.2:1
Abrupt Drift	100,000	24,557	0	20	N/A	4.2:1
Reoccurring Drift	100,000	23,952	0	20	N/A	4.2:1

B. Experiments on Effectiveness

As the increased research attention focuses on imbalanced learning, a number of evaluation metrics have been used to properly assess the effectiveness of classification with skewed distribution [5]. Among them, accuracy and recall are two fundamental measures for evaluating the performance of a classifier. A variety of evaluation metrics, for example G-mean, have evolved based on these two measures from different

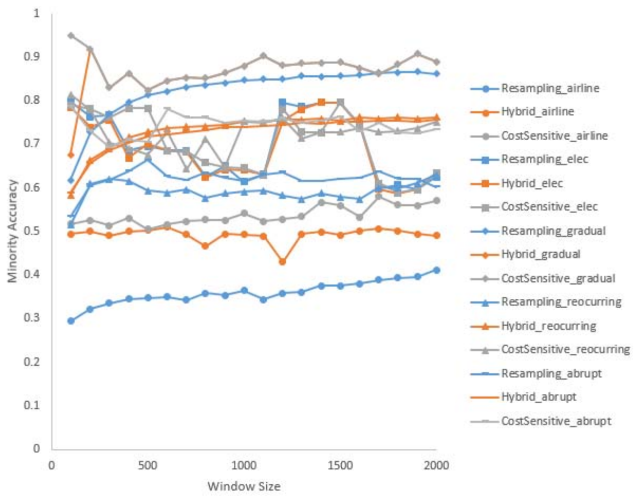

Fig. 2: Comparison of minority accuracy from Resampling, Hybrid and CostSensitive applied to five datasets.

perspectives. Although there is no measurement that specifically designed for imbalanced stream classification settings due to its relative paucity of research, the evaluation metrics of imbalanced learning can shed light on the performances of such methods. We therefore assess the effectiveness of our proposed framework using all the standardized evaluation metrics from imbalanced learning community, but only accuracy of minority, recall of minority as well as overall precision are shown for space efficiency consideration. As mentioned previously, although the rareness of minority class, it is usually the case under consideration on the study as it may carry important and useful knowledge, the accuracy and recall of minority provide evidence on the minority retrieval capabilities of a classifier. The overall accuracy helps us understand whether adjusting the learning bias from the majority towards the minority jeopardizes the accuracy of the majority class and to what degree.

For the parameter in resampling component, the stopping condition for resampling is set as the imbalance rate larger than 0.5, the performance threshold for updating component is also 0.5. In addition, the sizes of minority window and ensembled cost-sensitive classifier window are 100 and 10, respectively. The obtained results are shown from Figure 2 to Figure 4.

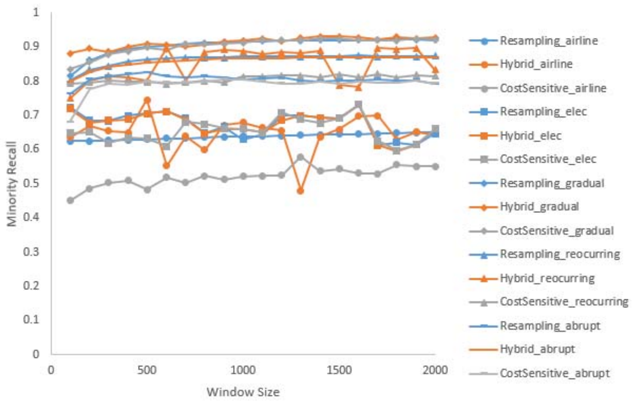

Fig. 3: Comparison of minority recall from Resampling, Hybrid and CostSensitive applied to five datasets.

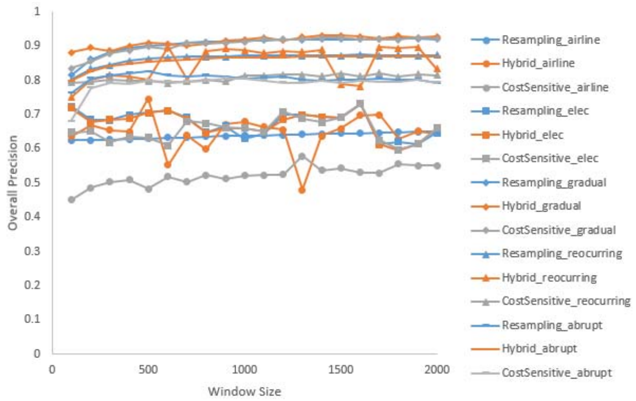

Fig. 4: Comparison of overall precision from Resampling, Hybrid and CostSensitive applied to five datasets.

The comparisons of minority accuracy shown in Figure 2 indicate on most of datasets, CostSensitive is able to identify minority instances with higher accuracy (grey lines tend to be above blue and orange lines in each dataset). It yields the highest minority accuracy of 0.95 on gradual drift dataset when the window size is 100. However, compared with Hybrid and Resampling, CostSensitive is inferior with respect to the recall of minority class (grey lines tend to be below blue and orange lines in each dataset). The worst recall 0.38 is also obtained by CostSensitive for the airline dataset, the window size is again 100. CostSensitive incorporates a high penalty for underrepresented set of instances. This makes classifiers pay more attention to minority class and can incur reluctance in predicting underrepresented instances due to their high penalty. On the contrary, Hybrid pre-reduces imbalance rate, the whole retrieval process is merit with exactness and completeness. What's more, in Figure 3 and Figure 4, Hybrid (blue lines) are most likely above Resampling (orange lines) in terms of each dataset, this verifies Hybrid's effectiveness, which allows it to be applicable in imbalanced stream classification application.

As shown in Figure 4, except for airline dataset, all predictors achieve good accuracy results in imbalance streaming classification settings. One possible reason for the not as good performance of airline dataset is the overfit results from its nominal attributes with a large number of classes [23] (e.g., the feature *AirportFrom* contains 293 distinct classes, etc.). Hybrid has the best overall accuracy for dataset gradual drift when window size is 1,500, with an accuracy held at 0.93. The results depict Hybrid manages the trade-off performance on minority class and majority class. All predictors require a certain size of sliding window in order to arrive at the best overall prediction. One possible reason for this is if the number of instances in the current sliding window is limited,

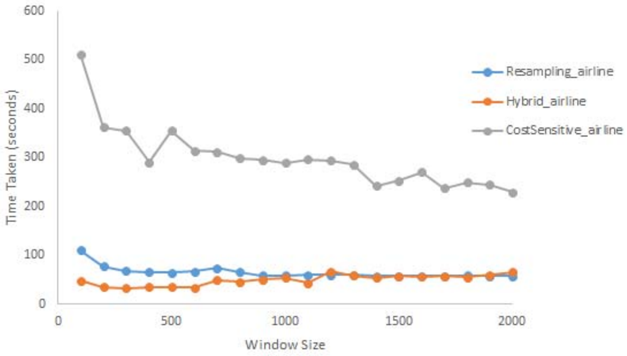

Fig. 5: Execution time in seconds for processing window size from 100 to 2000 applied to Airline dataset.

the information they contain is insufficient in return. However, the optimal window size differs in terms of different datasets and classifiers.

C. Experiments on Execution Efficiency

In addition to effectiveness, data stream frameworks are often employed in time-critical or resource constrained situations, research like that in [21] even explores the topic specifically. We gauge the efficiency of our proposed framework with the time required for processing each data stream with different window sizes.

Figure 5 to Figure 8 show the execution times reported in seconds on each dataset for Resampling, Hybrid and CostSensitive, and it is clear that Hybrid runs faster than Resampling and CostSensitive for most scenarios (the execution times of different datasets are shown in separate figures due to their varying time scale). When the size of sliding window is fixed, running time mainly dominant by the time for training and classifying. While classification time is generally equal, training time is responsible for the main execution time difference. The training process of CostSensitive involves relabeling each training example with the estimated optimal class, which is estimated from the ensemble classifier that learned from multiple bootstrap replicates of the training set. This incurs a relatively high cost in streaming data settings. In contrast, Resampling only considers the misclassified instance through time, which is available from the previous classification step. Hybrid applies CostSensitive to produce more efficient results after resampling. This is meaningful as aside from any performance requirement or resource constrains, data stream framework has an essential prerequisite that the classification task have a throughput at least equal to the rate at which instances or batches arrive.

VI. Conclusions

Although a number of studies have developed approaches to confront with non-stationary learning problem or class imbalance issue independently, the combined research of addressing these two problems concurrently has been underexplored. In this paper, we propose a learning framework that

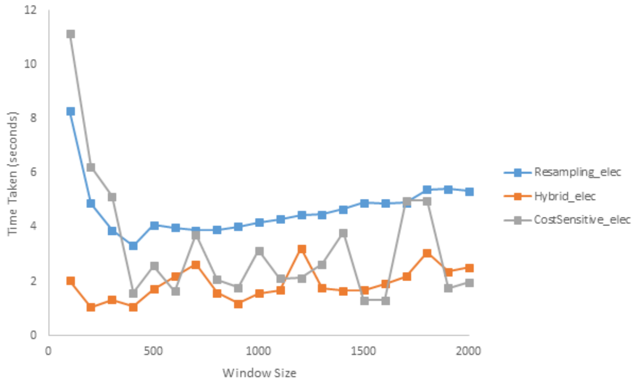

Fig. 6: Execution time in seconds for processing window size from 100 to 2000 applied to Electricity dataset.

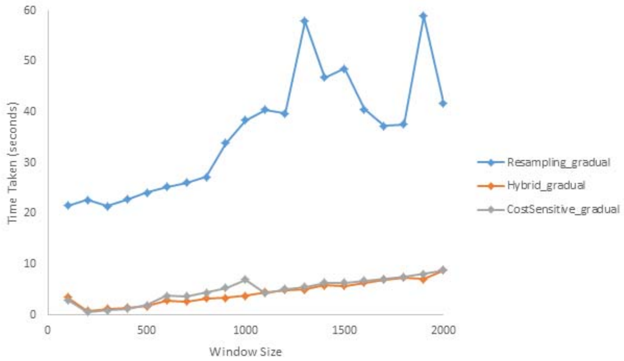

Fig. 7: Execution time in seconds for processing window size from 100 to 2000 applied to simulated Gradual Concept drift dataset.

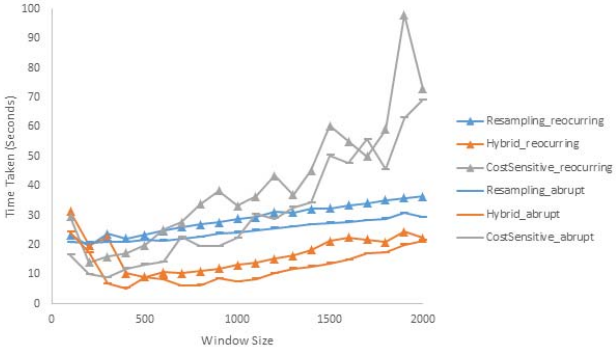

Fig. 8: Execution time in seconds for processing window size from 100 to 2000 applied to simulated Reoccuring Concept Drift and Abrupt Concept Drift datasets.

simultaneously deals with class imbalance and concept drift in imbalanced data streams, which have roots in a wide spectrum of real-world applications. The framework further incorporates cost sensitive solution according to the reduced imbalance rate. As the first work that introduces cost sensitive method in streaming data settings, we believe that the efficiency indicated by experimental results herein shows resampling in conjunction with cost sensitive is an excellent way to ensure a throughput for data stream framework. The results of the real-world dataset and synthetic datasets also suggest our proposed framework is an effective predictor of data streams. In the future, we plan to extend our work in a parallel manner for a better computation complexity reduction.

References

[1] J. Wang, et al. Wearable Sensor Based Human Posture Recognition. In *Proceedings of the IEEE International Conference on Big Data*, pages 432-438, 2016.

[2] P. Parveen , R.Z. Weger , B. Thuraisingham , K. Hamlen , and L. Khan. Supervised Learning for Insider Threat Detection Using Stream Mining. In *Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 1032-1039. IEEE Computer Society, 2011.

[3] R. Saia, L. Boratto and S. Carta. Multiple Behavioral Models: A Divide and Conquer Strategy to Fraud Detection in Financial Data Streams. In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 496-503, 2015.

[4] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy and A. Bouchachia. A Survey on Concept Drift Adaptation. *ACM Computing Surveys*, 46(4):441–4437, 2014.

[5] H. He and E.A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263-1284, 2009.

[6] T.R. Hoens, R. Polikar, N.V. Chawla. Learning from Streaming Data with Concept Drift and Imbalance: an Overview. *Progress in Artificial Intelligence*, 1:89-101, 2012.

[7] J. Gama. Knowledge Discovery from Data Streams (1st ed.). *Chapman & Hall/CRC*, 2010.

[8] P. Domingos and G. Hulten. Mining High-speed Data Streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 71–80. ACM, 2000.

[9] M. M. Lazarescu, S. Venkatesh and H.H. Bui. Using Multiple Windows to Track Concept Drift. *Intelligent Data Analysis*, 8(1): 29-59, 2004.

[10] W.N. Street and Y.S. Kim. A Streaming Ensemble Algorithm (SEA) for Large-scale Classification. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 377-382, ACM, 2001.

[11] N. V. C. et. al. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[12] J. Stefanowski. Dealing with Data Difficulty Factors While Learning from Imbalanced Data. *Challenges in Computational Statistics and Data Mining*, 605:333-363, Springer, 2016.

[13] P. Domingos. MetaCost: A General Method for Making Classifiers Cost-sensitive. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 155-164, 1999.

[14] N. Japkowicz, C. Myers and M. Gluck. A Novelty Detection Approach to Classification. In *Proceedings of the 14th international joint conference on Artificial intelligence*, 1:518-523, 1995.

[15] S. Wang, Z. Li, W.Chao, and Q. Cao. Applying Adaptive Oversampling Technique based on Data Density and Cost-sensitive SVM to Imbalanced Learning. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1-8, 2012.

[16] M. Woniak, M. Grana and E. Corchado. A Survey of Multiple Classifier Systems as Hybrid Systems. *Information Fusion*, 16(1):3-17, 2014.

[17] R. N. Lichtenwalter and N. V. Chawla Adaptive Methods for Classification in Arbitrarily Imbalanced and Drifting Data Streams. In *New Frontiers in Applied Data Mining*, 5669:53-75, 2010.

[18] D. Zhang, H. Shen, T. Hui, Y. Li, J. Wu and Y. Sang. A Selectively Retrain Approach Based on Clustering to Classify Concept-Drifting Data Streams with Skewed Distribution. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 413-424, 2014.

[19] T. Jo and N. Japkowicz. Class Imbalances Versus Small Disjuncts. *ACM SIGKDD Explorations Newsletter*, 6(1):40-49, 2004.

[20] I. Žliobaitė. How Good is the Electricity Benchmark for Evaluating Concept Drift Adaptation. *arXiv*:1301.3524, 2013.

[21] P. D. Haghighi, A. Zaslavsky, S. Krishnaswamy, M.M. Gaber and S. Loke. Context-aware Ddaptive Data Stream Mining. *Intelligent Data Analysis - Knowledge Discovery from Data Streams*, 13(3):423-434, 2009.

[22] M. G. Kelly, D. J. Hand and N. M. Adams. The Impact of Changing Populations on Classifier Performance. In *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM pages 367-371, 1999.

[23] G. James, D. Witten, T. Hastie and R. Tibshirani. An Introduction to Statistical Learning: With Applications in R. *Springer Publishing Company, Incorporated*, 2014.

[24] A. Bifet, G. Holmes, R. Kirkby and B. Pfahringer. Data Stream Mining: A Practical Approach. *Tech. rep. University of Waikato*, 2011.