Classification of Imbalanced Data Streams with Concept Drift: a Literature Review

R. Anderson, K. Chen and S. Jin

***Abstract*—We review the literature available around classification of imbalanced data in a streaming environment. First, we introduce the particular problems faced within streaming environments with class imbalance and concept drift. Through examining current research in the field, we explore the important considerations with performing analysis in this environment: sampling techniques; approaches to detecting concept drift; base learners for building models to classify unseen data; and methods of utilising ensembles to improve classification accuracy. We provide a discussion of measures that can be used to evaluate an approach, with specific reference to classifying rare classes. Finally, we use recent research as examples to explore the challenges of selecting a suitable approach.**

I. Introduction

In recent years, as computing power has increased dramatically, it has become possible to store and process large volumes of data. Such data sets, which continuously and rapidly grow over time, are referred to as data streams. Mining data streams brings challenges to traditional data mining techniques, as many traditional data mining techniques cannot be simply applied to data streams as: 1. Traditional data mining techniques need to process the data multiple times, which is difficult and costly with real-time data. 2. Data streams' 'temporal locality'. Their underlying distributions are more prone to change (concept drift), which makes it more difficult for traditional data mining techniques to adapt. Classification is much more difficult as the class distribution changes over time.

Furthermore, it's more challenging to improve the overall accuracy of classification if the classes' distribution are imbalanced, which is especially important for many applications in some fields, such as fraud detection and network intrusion detection.

When data comes in the form of streams, it is impossible to fit the entire data into machine's memory for processing, hence only online processing is feasible for mining data streams. Online processing refers to the predictive models being trained incrementally. However, this only solves the computational issue. In a dynamically changing environment, concept drift refers to the underlying class distribution of data stream potentially changing over time. A good example is online shopping customer preferences. To solve the problem of concept drift, predictive models needs to be regularly updated, which is also called online adaptive learning. This approach explicitly employs extra techniques to handle concept drift while keep the same base learning algorithms. However, it is also possible to modify the base learning algorithms to handle concept drift, such as Hellinger Distance Decision Tree(HDDT[1].) There are two strategies to update the learning model: 1. update the model at regular intervals without considering whether the concept has drifted; 2. detect a concept change before updating the model. In this survey, we mainly discuss the second approach. Detailed review of well known change detection algorithms are in in section 3.

An imbalanced data set has a skewed class distribution, with usually one majority class distribution, and one or more minority class distribution. The problem has been well researched, but recent researches focus on applying approaches to data streams. Three different types of approach are: 1. data level approaches, applying sampling techniques, which simply modify the data set in order to rebalance the class distribution; 2. algorithm level approaches, which are adapted to deal with identifying minority classes, improve classificationa accuracy; and 3. cost sensitive approaches, with cost matrices to weight the cost of misclassifying a class instance. Among them, data level approaches are mostly used in the context of data streams. Stream Ensemble Framework[2] is a good example of this. Many data level techniques for handling concept drift are combined with ensemble techniques to improve accuracy, as detailed in section 3, section 5 and section 7.

This literature survey gives an overview on handling concept drift with imbalanced data streams from different angles such as: sampling techniques; concept drift detection; base learning algorithms; ensemble techniques; and evaluation measures used for approaches that classify these streams.

The survey is organized as follows. Section 2 presents several sampling methods that are commonly used in data stream mining and imbalanced problem. Section 3 presents several common drift detection algorithms. Section 4 discusses several popular classification learning algorithms that are commonly used for streams, with a focus on decision trees. Section 5 discusses ensemble techniques. Section 6 discusses evaluating measures and methodologies that are used for skewed concept-drifted data stream classification. Section 7 discusses these sections, examining examples to show how the different aspects above can combine to provide suitable analysis. Section 8 discusses valuable future contributions to be made to the file, and Section 9 concludes the survey.

II. Sampling methods

There are two categories of sampling techniques that can be used for concept-drifting stream mining. One category aims to keep the training set small for fast processing, such as Sliding Windows and Reservoir Sampling [3]. The other category seeks to improve performance on the imbalanced data set. SMOTE [4] (Synthetic Minority Over-sampling Technique), SERA [5] (Selectively Recursive Approach) framework, MuSeRA [6] (Multiple Selectively Recursive Approach), REA [7] (Recursive Ensemble Approach) fall within this category. Balancing training data is the most straight-forward approach for imbalanced concept-drifting data stream mining. Many sampling algorithms improve accuracy by under-sampling the majority class or over-sampling the minority class.

In [2], Gao proved that the sampling technique can reduce error by collecting positive instances and keeping them in the training set. In SMOTE (Synthetic Minority Over-sampling Technique), the minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbours, rather than by over-sampling with replacement. SMOTE forces decision region of minority classes to become more general. SMOTE gives better classification accuracy in terms of ROC. SERA, MuSeRA, REA take a similar approach: either over-sampling, under-sampling or both. Neighbourhood Cleaning Rule (NCL[8]) uses the Wilson's Edited Nearest Neighbor Rule (ENN[8]) to remove majority class instances. In [9], the authors suggested using classifier ensembles with sampling techniques to improve classification performance. Sampling techniques are mainly used at the data level to handle imbalance problem, which has an advantages of not changing the learning algorithm. However, the data level approach suffers the problems from resource and computation. In contrast, algorithm level approach doesn't need sampling as a pre-processing step, thus doesn't have such problems.

III. Concept drift detection

Drift detection methods provide feedback to the learner by detecting when concept drifts have occurred. These methods may be independent of the learning model, or structurally embedded inside a classifying algorithm such as RCD [10]. When a concept drift has been detected by the detector, the learner is modified or retrained on some set of instances. This allows the learner to adapt and respond to changes which may lead to a higher classification accuracy compared to blind or non-adaptive approaches. Gama et al suggest one advantage of this approach is that it provides additional information about the way data was generated [11]. However this may create more false positives - particularly problematic for noisy data. A variety of change detection methods have been proposed - earlier work focused on sequential analysis techniques, but adaptive windowing and statistical methods are also popular approaches.

The CUSUM method is a sequential analysis technique that detects changes based on evaluating the cumulative residue of the learner and testing if there is a significant departure from zero [11]. When the cumulative value is greater than some threshold (λ) then it indicates a drift has occurred. The Page-Hinkley test is a variation on the CUSUM method that is used to detect sudden shifts in Gaussian signals [12]. An important application of this method is in signal processing. It records a minimum cumulative value in addition to the current cumulative value. An advantage of using the Page-Hinkley and CUSUM methods is that they are both memoryless.

The drift detection method (DDM) proposed by Gama et al [13] records the current mean error rate (p_i), and error standard deviation (s_i) as well as the minimum values $(p_{min},\ s_{min})$ as each new instance is seen [13]. A normal distribution is used to set the thresholds as the number of errors approximates a Bernoulli distribution for samples sizes greater than 30. This means that there is at least a delay of 30 errors until a drift is detected [14]. The warning threshold is defined as $p_{min} + 2 \times s_{min}$ and is updated with the arrival of new instances. A warning is raised when the current value of pi + si is greater or equal to the warning threshold and subsequent instances are stored in a warning window. If the error rate surpasses the drift threshold of $p_{min} + 3 \times s_{min}$, then the learner is retrained on instances stored in the warning window and the minimum values $(p_{min} and s_{min})$ are reset. An analysis done by Goncalves et al [15] on real and synthetic datasets shows that DDM performed most favourably on datasets affected by gradual drifts based on accuracies and the average ranking. The Early Drift Detection method (EDDM) is a variation on the drift detection method proposed by Gama et al [13] and was designed to handle gradual concept drifts [16]. It differs in that it uses the distance of the errors rather than the error rate as a measure of change. The main strength of this algorithm is the ability to detect slow gradual shifts [10]. On the other hand it may be less resistant to noise, and is much less effective for detecting sudden shifts. Hence it may have a high false alarm and miss detection rate for these datasets (as shown in Goncalves et als experiments [10] using datasets with abrupt drifts). Similar to DDM, this method may have a delay in drift detection due to the assumptions of the normal distribution approximation. Other algorithms like FLORA [17] use the overall accuracy of the learner as the measure of change, a large decrease in accuracy indicating the occurrence of a concept drift.

ADWIN is a popular change detection method based on comparing data distributions from two different windows via the use of an adaptive sliding window [18]. The detection method works by finding two sub-windows of the window W that have significantly different means. When two significantly different sub-windows are found the algorithm indicates a concept drift has occurred, and the window size is decreased by dropping the older sub-window. Otherwise the window size is enlarged when no drift is detected. As the window grows when no shift is detected, there is no upper limit to the window size during long periods of stability - this leads to an unbounded memory requirement (this problem is remedied in a later version of the algorithm). One limitation of the ADWIN approach is that it may require more memory than sequential or statistical approaches as it has a memory complexity of O(logW) - O(W) compared to constant time O(1) for the other approaches [11]. However it may give more precise information about the location of the concept drift [11]. In addition to this, it has strict guarantees on the rate of false positives and false negatives [15]. The time complexity of the algorithm is also higher than the other approaches but both experiments by Goncalves [15] and Gama [18] show that the runtime is lower than all other algorithms used in the studies.

The entropy-based approach also uses sliding windows and is based on a modified version of Shannon's entropy [19]. It uses a sliding 2 window approach, adapting the window size when a shift is detected. When a shift occurs window size is set to the minimum (forgets old instances), and grows with each new instance until the upper limit is reached.

Although there has been much work done on drift detection techniques there are few studies that attempt to address the class imbalance problem by modification of the drift detection method. Some of the more common approaches for dealing with imbalanced classes are the use of ensembles such as bagging or boosting, and sampling methods.

The PerfSim algorithm first trains the learner on a dataset, and produces a confusion matrix that summarises the performance of the learner [20]. After receiving a second dataset (batch of instances) it tests the learner on the new dataset and also produces a summary matrix. The summary statistics are converted into vectors and are compared using a similarity measure (cosine similarity). This may decrease the effect of class imbalance in comparison to methods that only rely on the overall accuracy where the contributions from the majority classes outweigh the minority classes. This method is independent of the learning algorithm and may be paired with any learner [20]. Antwi et al show that this method does not detect false drifts and performs well when evaluated against other current methods (VC and MMD) [20].

The DDM-OCI method is based on the DDM method proposed by Gama, but specifically attempts to deal with imbalanced classes [14]. This technique uses the recall of the minority class as the measure of change. The warning and drift thresholds are set in a similar manner to DDM and EDDM. The DDM-OCI algorithm has been shown to effectively detect drift in imbalanced datasets, but false alarm rates negatively affect the performance [14]. This causes a delay in detecting concept drift as the model and minimum values are reset [14].

IV. Base learners

Supervised learning requires models to be trained from labelled data. When model-building, we need an approach that can use explanatory variables and classes in a training set of data to create a set of rules for classifying unseen instances. This approach is our base learner: our tool that will build our classifier to apply to unseen

instances of our data [21].

In a streaming environment, however, we can only make one pass through our data. Concept drift requires a learner that can be run at regular intervals, so as to understand the current distribution that our instances fall into. This requires a trade-off between time, accuracy and memory. A successful learner will be sufficiently accurate while running quickly and with manageable memory overhead. These requirements are much more important than in a static environment, where we have a finite amount of data and time to process it.

Decision trees have particular features which make them a good fit for classifying streams with concept drift. The C4.5 decision tree, and its predecessor, the ID3 tree, are very popular within the research community [22]. Their appeal lies in their simplicity at every node, a binary decision is made related to one of the explanatory variables, resulting in a divide-and-conquer approach which results in fast classification and low overhead for the model. This leads to some clear drawbacks; only rectangular regions can be used to discriminate between classes in the feature space for instance. Splits are made through information gain measures, seeking to maximise the change in impurity of the nodes with each split. These trees will naturally overfit, if fully grown, but can be pruned to achieve a desired level of fit to the data. Their adaptability and speed make them a natural fit in classification of streams.

Domingos and Hulten [23] highlight a major concern with C4.5, ID3 and CART: they assume that all training examples can be held in memory, and are limited in the examples they can consider. Other trees such as SPRINT and SLIQ have attempted to resolve this issue by using a window to scan large datasets sequentially, but this is very slow when building complex trees, and still requires all data to reside in disk-space. They propose the Very Fast Decision Tree (VFDT), which requires that data items are only read once, in a small and constant time. They utilise Hoeffding Bounds [24] to guarantee that the statistically best attribute to divide a node on can very commonly be found using a small number of examples of instances as could be with infinite. They propose the Hoeffding tree, which regularly checks whether splitting a node on the best attribute will lead to an improvement in a selected measure (information gain or Gini) beyond a set threshold. This allows it to continue to adapt, even with infinite data, without sacrificing significant quality nor having burdensome memory requirements. Parameters need to be set by the user: the minimum number of examples before reviewing whether to split a node; the threshold to allow a split; and whether the algorithm can rescan prior examples. Even if these parameters are well selected by the user, there is a danger that they will lead to poor performance in environments featuring significant concept drift or imbalanced datasets.

Recognising the issues posed by concept drift, Hulten et al. [25] proposed the Concept-Adapting Very Fast Decision Tree (CVFDT), an adaptation on the VFDT to give it the adaptability to handle changing data. Through adopting a sliding window approach, the CVFDT increments counts of recently seen data while reducing counts of older data. This should have no effect when there is no concept drift, but where there is drift, prior splits will no longer pass the Hoeffding tests that deemed them worthwhile. At this point, an alternate subtree is grown by the algorithm from that node. If it accurately classifies new data better than the existing subtree, the existing subtree is replaced. Overall, this requires additional memory to keep alternate subtrees in memory, and summary statistics of split quality at each node. However, these memory requirements are still constant with the data items received. CVFDTs cannot utilise old sub-trees which are discarded, which could provide potential performance improvements where concepts drifts can revert back to prior probability distributions.

Decision trees generally have trouble accurately classifying rare classes. Consider a binary class problem: if a very large majority belongs to one class, often a classifier achieves best performance by classifying all instances as that class. Single classifiers often need to be combined in ensemble to classify rare classes in an imbalanced data set well. However recent research has made it possible for some classifiers to perform comparably to ensembles. Lyon et al [26] propose an improvement on the VFDT by introducing a new criterion for splitting: Hellinger distance. This measure seeks the attributes which are most disjoint in the data, and seeks to create tree splits based on these features. Instead of prioritising information gain, this tree prioritises class discrimination. This will often lead to a smaller portion correctly classified, but should amplify the number of the rare class successfully classified. Calculating Hellinger distance can be costly, as it scales to the number of classes and needs to discretize continuous classes into bins, leading to further multiplicative scaling. Lyon et al propose the HD-VFDT which uses the splitting criterion above, and the Gaussian Hellinger Very Fast Decision Tree, which assumes normality of the distribution to simplify calculation of the Hellinger Distance. However, instances in a data stream are not independent, as time affects their distribution, and so this could lead to flawed results in some circumstances. Experimental analysis of this approach showed statistically significant improvements using these methods over simple Hoeffding Trees where the minority class makes up 1% or less of the total data. They do not evaluate the approach thoroughly with regard to time nor memory use. Specifically, it would be interesting to measure the difference in accuracy, speed and time required between the two variants of the Hellinger tree proposed.

Lichtenwalter and Chawla [27] have found potential in using Hellinger trees in environments with concept drift. By weighting the Hellinger distance with Information Gain to measure distance between datasets, a learner can decide whether model learnt on previous data is valid on current data. Dal Pazzolo et al [28] show, using the HDDT proposed by Cieslak and Chawla [29] and C4.5 tree, that trees with this weighting generally outperformed trees without the weighting in class-imbalanced environments with drift. These papers do not address the multi-class problem, which would be a natural extension of this research.

V. Ensemble approaches

Ensembles consist of many components (typically different models or learners) which are combined by voting or averaging the outcomes from each component. Ensemble methods can often improve the accuracy of a classifying algorithm, and may be easier to scale and parallelize than single learners [30]. The main drawback is the increased memory and time requirement due to the extra time required to update and maintain the ensemble.

The Streaming Ensemble Algorithm (SEA) proposed by Street et al [31] uses an ensemble of decision trees to deal with concept drift. The data is partitioned into batches, with a decision tree being trained on each batch. The decision is then made based on an unweighted voting scheme. A quality measure is used for deciding when a new tree should replace an existing learner in the ensemble. In contrast to the unweighted ensemble learning in SEA presented by Street et al [31], Kolter and Maloof [32] suggest weighted base

learners may be a good approach for handling concept drift.

Bagging and boosting are two ensemble techniques that have been shown to have better performance than the individual models by Bauer and Kohavi, 1999 as cited by [33]. However these methods require repeated sampling of training data making them infeasible to be directly applied to the streaming environment [31]. Oza et al extend this technique to an online setting by developing online bagging and boosting methods [33]. Specifically they use a poisson sampling (with $\lambda = 1$) method to approximate the weighting of training data achieved by bagging [33]. Their online boosting method also uses poisson sampling but increases the rate when a base learner misclassifies an example.

Bifet et al [30] developed bagging and boosting methods for ADWIN and adaptive size Hoeffding trees based on Oza and Russells ensemble learning algorithms [33]. They showed that while bagging techniques greatly improves the accuracy (bagged ADWIN and ASHT had the highest accuracy for most datasets), there is a trade-off with the amount of time and memory used all bagging methods required more memory than non-bagged methods [30]. An experiment by Oza et al [33] comparing boosting and bagging methods on decision trees showed boosting methods (AdaBoost, and Online Boosting) performed as well as bagging methods (Online Bagging, and Bagging) based on the accuracy of the algorithms. In addition to this Scholz and Klinkenberg [34] show that employing a boosting method for training ensembles may adapt quickly to drifts with low computational overhead.

A number of ensembles have been introduced to address the class imbalance problem. For example Learn++ NSE as proposed by Ditzler et al [35], Learn++ SMOTE, and [2]. The Learn++ NSE algorithm increases the performance on imbalanced datasets, but has difficulty recovering from sudden concept drift. The framework outlined by Gao et al [2] focuses on a two class problem, and maintains a balanced dataset by retaining positive examples and under-sampling negative examples. They show that their approach reduces the mean square error on skewed datasets (0.9 to 0.1) and the training time required grows much slower than the single model learner [2].

VI. Evaluating approaches

Both datasets from influential institutions, such as the UCI repository, and synthetic datasets are often chosen to evaluate classifiers' performance. In order to form a data stream with a skewed distribution, a minority class and a majority class are chosen from a dataset to create the imbalanced relationship, with the remaining classes removed. The data will then be partitioned into chunks with the skewed distribution to be evaluated over time, to emulate the data stream.

Efficiency and accuracy are two major factors for evaluating learning algorithms' performance on a data stream. Efficiency often refers to time overhead. In this sense, sampling techniques often perform worse than other techniques (except when undersampling). In terms of accuracy, two measures that can be used include: (1) probability estimation accuracy (2) classification accuracy. In [2], Gao suggests that when using Mean Squared Error to measure the quality of a probability estimation for a rare class, low Mean Squared Error is desired. A common measurement for classification accuracy is classification error rate, but this measurement is undesirable for imbalanced data streams because the rare minority class doesn't have a significant impact on the classification error rate. Instead, three other measurement are typically used: Precision, Recall and False Alarm Rate. Gao suggested the ROC curve can show the trade-off between Precision and False Alarm Rate. A good ROC has big Area Under ROC Curve (AUROC): the closer the curve hugs the top-left corner of the plot, the better. Another similar method is the recall-precision plot. Gao showed that by employing the Sampling and Ensemble techniques (SE[2]), probability accuracy (MSE) and ROC measurement are both significantly improved. In [11], Gama discusses several other performance evaluation metrics that can be used: Sensitivity, Specificity and Kappa-statistics [11]. Kappa-statistics are especially useful when considering imbalanced datasets. All these performance metrics should be considered when comparing an analysis to a basic reference point or baseline analysis.

In [11], Gama states that alongside the evaluation metrics above, effectiveness of a change detection technique can also be measured. The suggested measures include: (1) Probability of true change detection. (2) Probability of false alarms. (3) Delay of detection.

Gama states that traditional cross-validation methods are too costly in data streams because of the velocity and volume of data received in a streaming environment. He suggests two procedures instead: (1) Holdout. This is widely used and the most useful method for validation, but it is not always possible, because of the speed of the data stream. (2) Interleaved Test-Then-Train or Prequential. This method makes full use of every instance, and has a smooth accuracy plot. (3) Controlled Permutations. This method is useful for sudden drift data stream by randomization.

VII. Discussion

In this paper thus far, we have outlined four important factors of streaming data analysis that are made more important by the presence of class imbalance and concept drift: sampling; concept drift detection; base learners; and ensemble approaches. We have also outlined various ways to evaluate the analysis performance. But how can we decide on a combination that best suits our purpose? Here, we examine research that has tested various approaches of on different measures, and summarise some of the lessons we can learn about designing our own analyses.

What analysis approach works best for a stream? Bifet et al. [36] examine analysis of very large data streams (1-10 mil. items) with drift present. The authors propose two new methods, with different approaches to ensembles and concept drift detection: bagging Adaptive Size Hoeffding Trees (where multiple Hoeffding Trees of different sizes are bagged, leading to increased diversity and increased performance); and bagged Hoeffding Trees, with ADWIN as a change detector (removing the worst classifier whenever change is detected). They compare their proposed two approaches against decision stumps, Naïve Bayes classifiers, and Hoeffding trees with DDM, EDDM and many others. They measure performance by time, memory and classification accuracy across datasets with various degrees of drift.

Their results demonstrate that there is no global best option. Bagged approaches provide better accuracy than other approaches at the cost of memory and time. However, using 5 trees rather than 10 almost doubles the speed of the approach without sacrificing much accuracy. The simplest approaches (e.g. Naïve Bayes) generally run in under a tenth of the time of the fastest, sacrificing anywhere between 5% and 30% classification accuracy compared to the more complex approaches, dependent on the drift in the underlying dataset. The memory requirements of some approaches such as

simple Hoeffding trees stay fairly constant over data items processed, while others such as bagged ASHT continually climb. The paper successfully demonstrates how its two proposed approaches perform very accurate analyses, but at a cost of time and memory.

This paper shows that before we decide on our approach, we need a clear understanding of the size of the stream, some understanding on whether underlying drift is present, and what constraints (time, memory and required accuracy) we have on our approach. This requires that we research and understand our constraints and data before implementing our approach, as a lot of time can be wasted through implementing insufficient approaches. Without regular (and costly) validation of stream classifiers, we will not fully understand how well our approach is working: using ensemble methods and including drift-detection helps to provide an approach that is robust in a changing environment.

We may then ask, do ensembles, enhanced sampling techniques and drift-detection always improve accuracy? Dal Pazzolo et al. [28] examine imbalanced datasets with varying levels of concept drift, and test various combinations of base learner (Hellinger Distance Decision Tree (HDDT) vs. C4.5), sampling method (baseline, undersampling, and two variations of oversampling the rare class), and ensemble approach (single model vs. an ensemble approach using the weighting of Hellinger Distance and Information Gain described previously). They evaluated their experiments in terms of time and area under the ROC curve.

Their results showed that undersampling and baseline samples performed faster than their oversampling methods. The HDDT trees led to slightly slower performance than the C4.5 trees. The fastest method (undersampling, C4.5 tree) was over twenty times faster than the slowest method (HDIG ensemble, with a HDDT tree and oversampling). AUROC measures were consistently improved by using their ensemble approach and the HDDT tree. The HDDT specifically made a large difference in the dataset with the highest imbalance ratio. However, this improvement was not significant when an ensemble approach was used. Overall, the baseline ensemble HDDT approach and the undersampled ensemble HDDT approach produced significantly better AUROC measures than the other approaches. Most interestingly, there was an interaction between learner type, sampling method, and ensemble approach, where a change in one factor would affect the efficacy of another. This set of results demonstrates two important points. First, without proper understanding of the methods we are using, we can lose time and accuracy through taking a complex approach, such as oversampling with ensemble HDDTs in this experiment. Second, when working with class imbalance, the degree and nature of imbalance will decide the relative improvement a particular approach provides.

Finally, how can we choose a suitable approach when we cannot predict the balance of our data? Zhang and Soda [37] examine binary-class classification in skewed data streams. They propose an approach of finding a balance between maximising overall classification accuracy with the balance of accuracy across classes, rather than maximising either. Their approach uses two different ensembles of classifiers (each a combination of Naïve Bayes, logistic regression and C4.5), each with a different method of analysis. The data is divided into batches, or fixed windows. The method used is decided by whether the data is detected to be skewed or not (a measure titled ”reliability”). Each method uses different sampling methods and ensemble voting approaches depending on whether it is for skewed or non-skewed data.

This decision-making process causes the approach to take longer than either individual ensemble, and achieves a lower total accuracy than one measure and class-balanced accuracy than the other. It does, however, provide a more robust and reliable result than the other two methods, regardless of whether the data has an underlying class imbalance or not. This approach doesn't explicitly address concept drift; if adapted to do so, though, it could provide an extremely robust approach to analysis of unseen data. This adaptability is valuable, as we expect our data distribution to change over time. Adaptive windowing provides similar value for streams with concept-drift. Adaptable approaches like these, however, generally come with a cost to performance.

VIII. Future Work

Analysis of imbalanced data streams with underlying concept drift is an exciting new area, with much to explore and develop. Here we discuss several areas that have come to our attention that would benefit from additional research.

Many approaches to imbalanced classes are limited to binary classes, and not multi-class problems. For example, Zhang and Soda?s two-ensemble approach uses evaluation measures and learners that cannot function beyond two classes. In the streaming environment, accommodating multiple-classes is expensive and a lot of work will be needed to adapt these approaches.

Performance comparisons between new approaches are few and far between, especially across dimensions explored in this paper. Research that compares performance of existing approaches in total accuracy, class-balanced accuracy, time and memory across different ensemble methods, sampling, learners and concept-drift detection will be valuable.

Finally, selecting an approach right now requires a deep understanding of the data stream. Adaptive approaches that can select powerful approaches to analysing unseen data will be invaluable. These must be acceptably cheap to perform in the worst-case and adaptive enough to adjust to changing underlying data streams.

IX. Conclusion

In this paper, we have examined analysis of data streams, with a focus on those with class-imbalance and underlying concept drift. We have discussed approaches to sampling, concept drift detection, base learners and ensemble approaches, and how they contribute to analyses. With each of these aspects, we have discussed important and influential techniques that are making an impact within the area.

We explored measures that can be used to evaluate our methods, and the specific considerations that need to be made with skewed data. Through discussing examples of these approaches put into practice, we have demonstrated the complexity of selecting an approach, and shown how we must understand the constraints and goals of our analysis before deciding on an approach.

Finally, we have proposed several areas of future work that have the potential to make a real contribution to this field.

REFERENCES

[1] D. Cieslak, T. Hoens, N. Chawla, and W. Kegelmeyer, "Hellinger distance decision trees are robust and skew-insensitive," *Data Mining and Knowledge Discovery*, vol. 24, no. 1, pp. 136–158, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10618-011-0222-1

[2] J. Gao, W. Fan, J. Han, and S. Y. Philip, "A general framework for mining concept-drifting data streams with skewed distributions." in *SDM*. SIAM, 2007, pp. 3–14.

[3] P. Vorburger and A. Bernstein, "Entropy-based concept shift detection," in *Data Mining, 2006. ICDM'06. Sixth International Conference on*. IEEE, 2006, pp. 1113–1118.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *arXiv preprint arXiv:1106.1813*, 2011.

[5] S. Chen and H. He, "Sera: selectively recursive approach towards nonstationary imbalanced stream data mining," in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. IEEE, 2009, pp. 522–529.

[6] S. Chen, H. He, K. Li, and S. Desai, "Musera: multiple selectively recursive approach towards imbalanced stream data mining," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*. IEEE, 2010, pp. 1–8.

[7] S. Chen and H. He, "Towards incremental learning of nonstationary imbalanced data stream: a multiple selectively recursive approach," *Evolving Systems*, vol. 2, no. 1, pp. 35–50, 2011.

[8] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[9] A. Godase and V. Attar, "Classifier ensemble for imbalanced data stream classification," in *Proceedings of the CUBE International Information Technology Conference*. ACM, 2012, pp. 284–289.

[10] P. M. G. Jr and R. S. M. de Barros, "RCD: A recurring concept drift framework," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1018 – 1025, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865513000494

[11] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.

[12] E. Page, "Continuous inspection schemes," *Biometrika*, pp. 100–115, 1954.

[13] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *In SBIA Brazilian Symposium on Artificial Intelligence*. Springer Verlag, 2004, pp. 286–295.

[14] S. Wang, L. Minku, D. Ghezzi, D. Caltabiano, P. Tino, and X. Yao, "Concept drift detection for online class imbalance learning," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, Aug 2013, pp. 1–10.

[15] P. M. G. Jr., S. G. de Carvalho Santos, R. S. Barros, and D. C. Vieira, "A comparative study on concept drift detectors," *Expert Systems with Applications*, vol. 41, no. 18, pp. 8144 – 8156, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417414004175

[16] M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno, "Early drift detection method," 2006.

[17] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine learning*, vol. 23, no. 1, pp. 69–101, 1996.

[18] A. Bifet and R. Gavald, *Learning from Time-Changing Data with Adaptive Windowing*, ch. 42, pp. 443–448. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611972771.42

[19] P. Vorburger and A. Bernstein, "Entropy-based concept shift detection," in *Data Mining, 2006. ICDM '06. Sixth International Conference on*, Dec 2006, pp. 1113–1118.

[20] D. Antwi, H. Viktor, and N. Japkowicz, "The PerfSim algorithm for concept drift detection in imbalanced data," in *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, Dec 2012, pp. 619–628.

[21] T. Hoens, R. Polikar, and N. Chawla, "Learning from streaming data with concept drift and imbalance: an overview," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 89–101, 2012. [Online]. Available: http://dx.doi.org/10.1007/s13748-011-0008-0

[22] S. L. Salzberg, "Book review: C4.5: Programs for machine learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993," *Mach. Learn.*, vol. 16, no. 3, pp. 235–240, Sep. 1994. [Online]. Available: http://dx.doi.org/10.1023/A:1022645310020

[23] 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*, ser. KDD '00. New York, NY, USA: ACM, 2000, pp. 71–80. [Online]. Available: http://doi.acm.org/10.1145/347090.347107

[24] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. pp. 13–30, 1963. [Online]. Available: http://www.jstor.org/stable/2282952

[25] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. New York, NY, USA: ACM, 2001, pp. 97–106. [Online]. Available: http://doi.acm.org/10.1145/502512.502529

[26] R. Lyon, J. Brooke, J. Knowles, and B. Stappers, "Hellinger distance trees for imbalanced streams," *arXiv preprint arXiv:1405.2278*, 2014.

[27] 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*. Springer, 2010, pp. 53–75.

[28] A. Dal Pozzolo, R. Johnson, O. Caelen, S. Waterschoot, N. V. Chawla, and G. Bontempi, "Using HDDT to avoid instances propagation in unbalanced and evolving data streams," 2014, in 2014 IEEE World Congress on Computational Intelligence.

[29] D. Cieslak and N. Chawla, "Learning decision trees for unbalanced data," in *Machine Learning and Knowledge Discovery in Databases*, ser. Lecture Notes in Computer Science, W. Daelemans, B. Goethals, and K. Morik, Eds. Springer Berlin Heidelberg, 2008, vol. 5211, pp. 241–256. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-87479-9_34

[30] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 139–148. [Online]. Available: http://doi.acm.org/10.1145/1557019.1557041

[31] W. N. Street and Y. 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*, ser. KDD '01. New York, NY, USA: ACM, 2001, pp. 377–382. [Online]. Available: http://doi.acm.org/10.1145/502512.502568

[32] J. Z. Kolter and M. A. Maloof, "Using additive expert ensembles to cope with concept drift," in *In Proceedings of the 22nd International Conference on Machine Learning (ICML-2005)*. ACM Press, 2005, pp. 449–456.

[33] N. C. Oza and S. Russell, "Online bagging and boosting," in *In Artificial Intelligence and Statistics 2001*. Morgan Kaufmann, 2001, pp. 105–112.

[34] M. Scholz and R. Klinkenberg, "An ensemble classifier for drifting concepts," in *In Proceedings of the Second International Workshop on Knowledge Discovery in Data Streams*, 2005, pp. 53–64.

[35] G. Ditzler and R. Polikar, "An ensemble based incremental learning framework for concept drift and class imbalance," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, July 2010, pp. 1–8.

[36] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 139–148.

[37] C. Zhang and P. Soda, "A double-ensemble approach for classifying skewed data streams," in *Advances in Knowledge Discovery and Data Mining*, ser. Lecture Notes in Computer Science, P.-N. Tan, S. Chawla, C. Ho, and J. Bailey, Eds. Springer Berlin Heidelberg, 2012, vol. 7301, pp. 254–265. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30217-6_22