THE INTERNATIONAL JOURNAL OF
SCIENCE & TECHNOLEDGE

Novel Class Dectection by Using Ensemble Classifier Framework

Shaik Abdullah G.
PG Student, Department of CSE, A.V.C. College of Engineering, Mayiladuthurai

Abstract:
Data stream classification poses many challenges to the data mining community. In this paper, we address four such major challenges, namely, infinite length, concept-drift, concept-evolution, and feature-evolution. Since a data stream is theoretically infinite in length, it is impractical to store and use all the historical data for training. Concept-drift is a common phenomenon in data streams, which occurs as a result of changes in the underlying concepts. Concept-evolution occurs as a result of new classes evolving in the stream. Feature-evolution is a frequently occurring process in many streams, such as text streams, in which new features (i.e., words or phrases) appear as the stream progresses. Most existing data stream classification techniques address only the first two challenges, and ignore the latter two. In this paper, we propose an ensemble classification framework, where each classifier is equipped with a novel class detector, to address concept-drift and concept-evolution. To address feature-evolution, we propose a feature set homogenization technique. We also enhance the novel class detection module by making it more adaptive to the evolving stream, and enabling it to detect more than one novel class at a time. Comparison with state-of-the-art data stream classification techniques establishes the effectiveness of the proposed approach.

Keywords: *Data mining community, data stream classification techniques, concept-drift and concept-evolution*

1. Introduction
Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

2. Objective of this Project
It is impractical to store and use all the historical data for training. The most obvious alternative is an incremental learning technique. Several incremental learners have been proposed to address this problem . In addition, concept-drift occurs in the stream when the underlying concepts of the stream change over time.

A variety of techniques have also been proposed in the literature for addressing concept-drift in data stream classification. However, there are two other significant characteristics of data streams, namely, concept-evolution and feature evolution, that are ignored by most of the existing techniques.

Concept-evolution occurs when new classes evolve in the data. For example, consider the problem of intrusion detection in a network traffic stream. If we consider each type of attack as a class label, then concept evolution occurs when a completely new kind of attack occurs in the traffic. Another example is the case of a text data stream, such as that occurring in a social network such as Twitter. In this case, new topics (classes) may frequently emerge in the underlying stream of text messages. The problem of concept-evolution is addressed in only a very limited way by the currently available data stream classification techniques. We investigate this problem in this paper, and propose improved solutions.

Our current work also addresses the feature-evolution problem in data streams, such as text streams, where new features (words) emerge and old features fade away. Masud et al. address the novel class detection problem in the presence of concept-drift and infinite length. In this technique, an ensemble of models is used to classify the unlabeled data, and detect novel classes. The novel class detection process consists of three steps. First, a decision boundary is built during training.

Second, test points falling outside the decision boundary are declared as outliers. Finally, the outliers are analyzed to see if there is enough cohesion among themselves (i.e., among the outliers) and separation from the existing class instances. But Masud et al. did not address the feature-evolution problem. The feature-evolution problem is addressed in, which also addressed the concept-evolution problem.

However, both and have two drawbacks. First, the false alarm rate (i.e., detection of existing classes as novel) is high for some data sets. Second, if there is more than one novel class, they are unable to distinguish among them. In this work, we propose a superior technique for both outlier detection and novel class detection to reduce both false alarm rate and increase detection rate. Our framework also allows for methods to distinguish among two or more novel classes.

We claim four major contributions in novel class detection for data streams. First, we propose a flexible decision boundary for outlier detection by allowing a slack space outside the decision boundary. This space is controlled by a threshold, and the threshold is adapted continuously to reduce the risk of false alarms and missed novel classes. Second, we apply a probabilistic approach to detect novel class instances using the discrete Gini Coefficient.

With this approach, we are able to distinguish different causes for the appearance of the outliers, namely, noise, concept-drift, or concept-evolution. We derive an analytical threshold for the Gini Coefficient that identifies the case where a novel class appears in the stream. We empirically show the effectiveness of this approach. Third, we apply a graph-based approach to detect the appearance of more than one novel class simultaneously, and separate the instances of one novel class from the others.

Finally, our proposed approach addresses the feature evolution problem on top of the enhancements discussed above. To the best of our knowledge, this is the first work that proposes these advanced techniques for novel class detection and classification in data streams and addresses feature-evolution. We apply our technique on a number of benchmark data streams including Twitter messages, and outperform the state-of-the-art classification and novel class detection techniques.

It designed to function as a multi-class classifier for concept-drifting data streams, detect novel classes, and distinguish recurring classes from novel classes. We keep an ensemble of size, and also keep an auxiliary ensemble where at most models per class are stored.

This auxiliary ensemble stores the classes in the form of classification models even after they disappear from the stream. Therefore, when a recurring class appears, it is detected by the auxiliary ensemble as recurrent. This approach greatly reduces false alarm rate as well as the overall error. If, however, a completely new class appears in the stream, it is detected as novel by the auxiliary ensemble as well. This is the first work that addresses the recurring class issue and concept-evolution in data streams. Our proposed solution, which uses an auxiliary ensemble for recurring class detection, reduces false alarm rates and overall classification error. Second, this technique can be applied to detect periodic classes, such as classes that appear weekly, monthly, or yearly. This will be useful for better predicting and profiling the characteristics of a data stream. Finally, we apply our technique on a number of real and synthetic datasets, and obtain superior performance over state-of-the-art techniques. A recurring class is a special and more common case of concept-evolution in data streams. It occurs when a class reappears after long disappearance from the stream. Recurring classes, when unaddressed, create several undesirable effects. First, they increase the false alarm rate because when they reappear, they may be falsely identified as novel, whereas such classes may observe normal representative behavior. Second, they also increase human effort, in cases where the output of the classification is used by human analyst.

In such cases, the analyst may have to spend extra effort in analyzing the afore-mentioned false alarms. Finally, additional computational effort is wasted in running a "novel class detection" module, which is costlier than regular "classification" process. In this approach, a fixed size ensemble is used classify the data stream and detect novel classes. When a novel class appears in the stream, it is soon added to the list of classes that the ensemble represents. However, the ensemble is periodically refined with new data, and therefore, if some class disappears for a long time that class is eventually dropped from the list.

2.1 Overview of Data Stream Classification

In which deserves attention, but has long assumed that the numbers of classes are fixed. However, in data streams, new classes may often appear. For example, a new kind of intrusion may appear in network traffic, or a new category of text may appear in a social text stream such as Twitter. When a new class emerges, traditional data stream classifiers misclassify the instances of the new class as one of the old classes. In other words, a traditional classifier is bound to misclassify any instance belonging to a new class, because the classifier has not been trained with that class. It is important to be able to proactively detect novel classes in data streams. For example, in an intrusion detection application, it is important to detect and raise alerts for novel intrusions as early as possible, in order to allow for early remedial action and minimization of damage.

A recurring class is a special and more common case of concept-evolution in data streams. It occurs when a class reappears after long disappearance from the stream. Recurring classes, when unaddressed, create several undesirable effects. First, they increase the false alarm rate because when they reappear, they may be falsely identified as novel, whereas such classes may observe normal representative behavior. Second, they also increase human effort, in cases where the output of the classification is used by human analyst. In such cases, the analyst may have to spend extra effort in analyzing the afore-mentioned false alarms. Finally, additional computational effort is wasted in running a "novel class detection" module, which is costlier than regular "classification" process. Novel class detection is major concept of concept evolution. In data stream classification assume that total no of classes is fixed but not be valid in a real streaming environment. When new class may evolve at any time.

Classifier ensembles are a common way of boosting classification accuracy. Due to their modularity, they also provide a natural way of adapting to change by modifying ensemble members. Ensemble algorithms are sets of single classifiers whose decisions are aggregated by a voting rule. The combined decision of many single classifiers is usually more accurate than that given by a single component. In this project we use the ensemble classifiers to mine evolving data streams.

When the class reappears, it is again detected as a novel class, although it should use with the class in order to provide a more accurate result. Another work in novel concept detection does not distinguish between novel class and recurrence class, i.e., it considers all classes as novel other than a prespecified "normal" class. Therefore, the problem remains, i.e., once a class has appeared in the stream, how we may "remember" this class as not novel when it appears again after a long absence from the stream.

Data stream classification has been a widely studied research problem in recent years. The dynamic and evolving nature of data streams requires efficient and effective techniques that are significantly different from static data classification techniques. Two of the most challenging and well-studied characteristics of data streams are its infinite length and concept-drift. Since a data stream is a fast and continuous phenomenon, it is assumed to have infinite length. Therefore, it is impractical to store and use all the historical data for training.

2.2 Incremental Learning Technique

Several incremental learners have been proposed to address this problem. In addition, concept-drift occurs in the stream when the underlying concepts of the stream change over time. However, there are two other significant characteristics of data streams, namely, concept-evolution and feature evolution, that are ignored by most of the existing techniques. Concept-evolution occurs when new classes evolve in the data. For example, consider the problem of intrusion detection in a network traffic stream. If we consider each type of attack as a class label, then concept evolution occurs when a completely new kind of attack occurs in the traffic. Another example is the case of a text data stream, such as that occurring in a social network such as Twitter.

In this case, new classes may frequently emerge in the underlying stream of text messages. The problem of concept-evolution is addressed in only a very limited way by the currently available data stream classification techniques. We investigate this problem in this paper, and propose improved solutions. Our current work also addresses the feature-evolution problem in data streams, such as text streams, where new features emerge and old features fade away address the novel class detection problem in the presence of concept-drift and infinite length. In this technique, an ensemble of models is used to classify the unlabeled data, and detect novel classes. The novel class detection process consists of three steps. First, a decision boundary is built during training. Second, test points falling outside the decision boundary are declared as outliers.

In this work, we propose a superior technique for both outlier detection and novel class detection to reduce both false alarm rate and increase detection rate. Our framework also allows for methods to distinguish among two or more novel classes. We claim four major contributions in novel class detection for data streams. First, we propose a flexible decision boundary for outlier detection by allowing a slack space outside the decision boundary. This space is controlled by a threshold, and the threshold is adapted continuously to reduce the risk of false alarms and missed novel classes. Second, we apply a probabilistic approach to detect novel class instances using the discrete Coefficient.

2.3 Training Phase

Once a new model is trained, it replaces one of the existing models in the ensemble. The candidate for replacement is chosen by evaluating each model on the latest training data, and selecting the model with the worst prediction error. This ensures that we have exactly L models in the ensemble at any given point of time. In this way, the infinite length problem is addressed because a constant amount of memory is required to store the ensemble.

2.4 Feature Space Conversion

It is obvious that the data streams that do not have any fixed feature space (such as text stream) will have different feature spaces for different models in the ensemble, since different sets of features would likely be selected for different chunks. Besides, the feature space of test instances is also likely to be different from the feature space of the classification models. Therefore, when we need to classify an instance, we need to come up with a homogeneous feature space for the model and the test instances.

There are three possible alternatives:

- Lossy fixed conversion (or Lossy-F conversion in short)
- Lossy local conversion (or Lossy-L conversion in short)
- Lossless homogenizing conversion (or Lossless in short).

2.5 Outlier Detection Using Adaptive Threshold

We allow a slack space beyond the surface of each hyper sphere. If any test instance falls within this slack space, then it is considered as existing class. This slack space is defined by a threshold, which is referred to as OUTTH. Therefore, we apply an adaptive technique to adjust the false alarm rate. First, we explain how to use the threshold, and then we will discuss how to adjust.

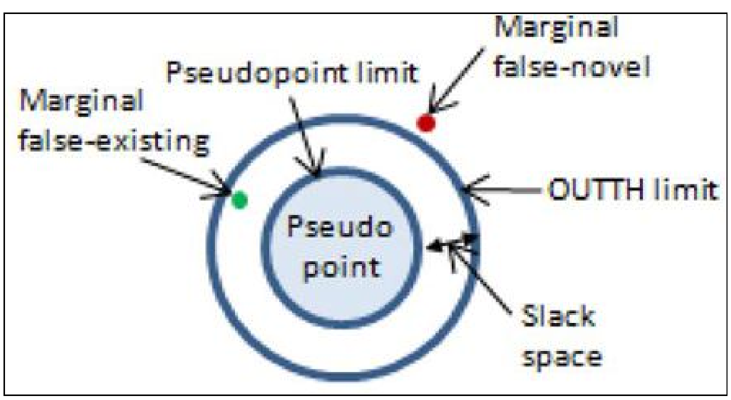

Figure 1: Illustrating Outlier Threshold

2.6 Novel Class Detection Using Gini Coefficient

After detecting the F-outlier instances using the OUTTH value discussed in the previous section, we compute the q- NSC(x) value for each F-outlier instance x using (1). If the q-NSC(x) value is negative, we remove x from consideration, i.e., x is regarded as an existing class instance. For the remaining F-outliers, q-NSC(.) is within the range [0, 1].Now, we compute a compound measure for each such Foutlier, called Novelty score or Nscore.

2.7 Simultaneous Multiple Novel Class Detection

The main idea in detecting multiple novel classes is to construct a graph, and identify the connected components in the graph. The number of connected components determines the number of novel classes. The basic assumption in determining the multiple novel classes follows property 1. For example, if there are two novel classes, then the separation among the different novel class instances should be higher than the cohesion among the same-class instances.

3. System Analysis

3.1 Existing System

In existing system use actminer applies an ensemble classification technique but used for limited labeled data problem and addressing the other three problem so reducing the cost. actminer is extends from mine class. Actminer integrates with four major problem concept drift, concept evolution, novel class detection, limited labeled data instances. But in this technique dynamic feature set problem and multi label classification in data stream classification. Based on clustering methods for collecting potential novel instances so memory is required to store. Another disadvantage is that using clustering method first find centroid. And also incremental so time overhead occurs. And also not possible classify streamed data continuously.

3.2 Drawbacks Of Existing System

- Memory overhead occurs.
- CPU overhead occurs
- Difficult to identify the misclassified novel classes.

3.3 Proposed System

Data Stream means continuous flow of data. Example of data stream includes computer network traffic, phone conversation, ATM transaction, and Web Searches and Sensor data. Data Stream Mining is a process of extracting knowledge structure from continuous, rapid data records. It can be considered as a subfield of data mining. Data Stream can be classified into online streams and offline streams. Online Data stream mining used in a number of real world applications, including network traffic monitoring, intrusion detection and credit card fraud detection. And offline data stream mining used in like generating report based on web log streams. Characteristics of data stream are continuous flow of data. Data size is extremely large and potentially infinite.

- Infinite Training Data-Can't store or use all historical data for training.
- Concept drift-Data changes over time. Historical training data built a model on those data which are outdated.
- Novel class-Novel class may appear over time.

Data stream have infinite length multi pass learning algorithm can not applicable as they would required infinite storage. Concept drift occurs when data changes over time. Another major problem is ignored by state of art data stream classification techniques which is concept evolution that means emergence of novel class. Assume that total no of classes is fixed. But in real data stream classification problems such as intrusion detection, text classification and fault detection Novel class may appear at any time in a stream. So all novel class instance go undetected until novel class manually detected by experts.

Novel class detection is major concept of concept evolution. In data stream classification assume that total no of classes is fixed but not be valid in a real streaming environment. In this project we proposed novel class detector to analyze the novel classes. Then use the outlier detection using adaptive threshold. We perform the novel class detection using Gini coefficient, and identify the simultaneous multiple novel class detection.

3.4 Advantages Of Proposed System

- An improved technique for outlier detection by defining a slack space outside the decision boundary of each classification model, and adaptively changing this slack space based on the characteristic of the evolving data
- Efficient method for evaluate the data streams using our proposed algorithm.

3.5 Proposed System Algorithm Details

The proposed novel class detection approach. The input to the algorithm is the ensemble M and the buffer Buffer holding the outliers instances. At first, we create K0 clusters using K-means with the instances in Buf (line 2), where K0 is proportional to K, the number of pseudo points per chunk (line 1).Efficient method for evaluate the data streams using our proposed algorithm.. Then each cluster is transformed into a pseudo point data structure, which stores the centroid, weight (number of data points in the cluster) and radius (distance between the centroid and the farthest data point in the cluster). Clustering is performed to speed up the computation of q-NSC value . If we compute q-NSC value for every F-outlier separately, it takes quadratic time in the number of the outliers. On the otherhand, if we compute the q-NSC value of the K0F-outlier pseudopoints (or O-pseudopoint), it takes constant time. The q-NSC value of a O-pseudopoint h is the approximate average of the q-NSC value of each instance in h. This is computed as follows: First, we define c; qðhÞ in terms of a Opseudopoint h.

3.5.1. Algorithm
Detect-Novel(M,Buf)
Input: M: Current ensemble of best L classifiers
Buf: Buffer temporarily holding F-outlier instances
Output: The novel class instances identified, if found
1: Ko (K_jBufj=S) //S ¼ chunk size K ¼ clusters per chunk
2: H K-means(Buf,K0) //create K0 O-pseudopoints
3: for each classifier Mi 2 M do
4: tp 0
5: for each cluster h 2 H do
6: h:sc q-NSC(h) //(equation 1)
7: if h:sc > 0 then
8: tp þ ¼ h:size //total instances in the cluster
9: for each instance x 2 h:cluster do x:sc max(x:sc,h:sc)
10: end if
11: end for
12: if tp > q then voteþþ
13: end for
14: if vote ¼¼ L then //found novel class, identify novel instances
15: Xnov all instance x with x:sc > 0
16: for all x 2 Xnov do
17: x:ns Nscore(x) //equation 2
18: if x:ns > Ginith then N list N list x
19: end for
20: Detect-Multinovel(N list)
21: endif

4. System Style

4.1. System Design

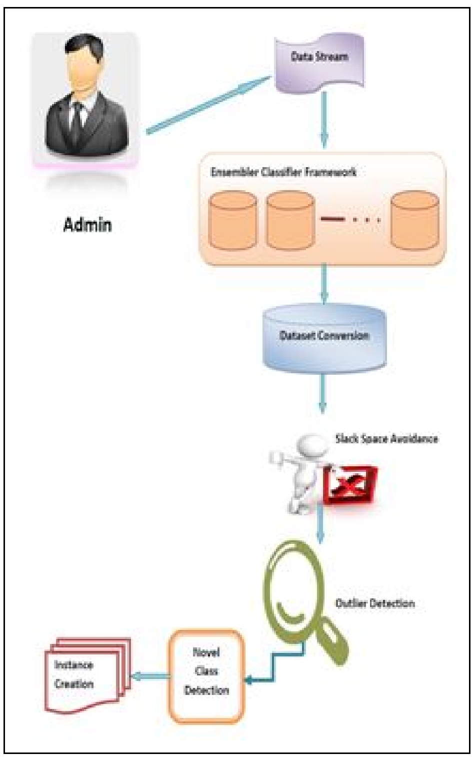

Figure 2: System Design

5. Modules Description

5.1. Upload Data Streams
A data stream is an ordered sequence of instances that in many applications of data stream mining can be read only once or a small number of times using limited computing and storage capabilities. Examples of data streams include computer network traffic, phone conversations, ATM transactions, web searches, and sensor data. Data stream mining can be considered a subfield of data mining, machine learning, and knowledge discovery. We can upload large volume of ordered data points, possibly infinite. Then it can be arrive continuously and fast changing type.

5.2. Ensemble Classifier
Supervised learning algorithms are commonly described as performing the task of searching through a hypothesis space to find a suitable hypothesis that will make good predictions with a particular problem. Even if the hypothesis space contains hypotheses that are very well-suited for a particular problem, it may be very difficult to find a good one. Ensembles combine multiple hypotheses to form a (hopefully) better hypothesis. In other words, an ensemble is a technique for combining many weak learners in an attempt to produce a strong learner. The term ensemble is usually reserved for methods that generate multiple hypotheses using the same base learner.
The broader term of multiple classifier systems also covers hybridization of hypotheses that are not induced by the same base learner models. Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a machine learning ensemble refers only to a concrete finite set of alternative models. Evaluating the prediction of an ensemble typically requires more computation than evaluating the prediction of a single model, so ensembles may be thought of as a way to compensate for poor learning algorithms by performing a lot of extra computation.

5.3. Decision Boundary Outlier Detection
The novel class detection process consists of three steps. First, a decision boundary is built during training. Second, test points falling outside the decision boundary are declared as outliers. Finally, the outliers are analyzed to see if there is enough cohesion among themselves and separation from the existing class instances. This space is controlled by a threshold, and the threshold is adapted continuously to reduce the risk of false alarms and missed novel classes. Second, we apply a probabilistic approach to detect novel class instances using the discrete Gini Coefficient. With this approach, we are able to distinguish different causes for the appearance of the outliers, namely, noise, concept-drift, or concept-evolution. We derive an analytical threshold for the Gini Coefficient that identifies the case where a novel class appears in the stream.

5.4. Novel Class Detector
We propose a superior technique for both outlier detection and novel class detection to reduce both false alarm rate and increase detection rate. Our framework also allows for methods to distinguish among two or more novel classes. When there are enough instances in the buffer, the novel class detection module is invoked. If a novel class is found, the instances of the novel class are tagged accordingly. Otherwise, the instances in the buffer are considered as an existing class and classified normally using the ensemble of models.

5.5. Misclassified Instances Detection
When data is come than according to that class will be created. But in that some data are misclassified. But no need to collect misclassifies instances. When we detect single potential instance then model is trained, classify that instance and create a new class. So when those types of instances come that then classify in that class.

5.6. Performance Evaluation
This module is used to evaluate the high false positive (false-novel class detection) rate and false negative (missed novel class detection) rates in some data sets.

6. Conclusion
The novel class detection is major task in data streams. Most of the novelty detection techniques either assume that there is no concept-drift, or build a model for a single "normal" class and consider all other classes as novel. But our approach is capable of detecting novel classes. In this project we proposed the Novel class detector and analyze the misclassified novel class. We can use the Novel class detector to enhance the novel class detection module by making it more adaptive to the evolving stream, and enabling it to detect more than one novel class at a time. In this paper we introduce new voting method to detect novel class using coefficient option tree in concept drifting data stream classification which builds a decision tree from data stream. Here we can train a model when potential novel instance is found. Not require to collect misclassified instances. So do not require to further classification. Timing and accuracy is improved.

7. Future Enhancement
An interesting future work would be to identify this special case more precisely to distinguish from the actual arrival of a novel class. An interesting and relevant question here is what will happen if one class split into several classes. If after split, they occupy the same feature space, meaning, the feature space they were covering before split is the same as the union of the feature spaces covered after split, none of the new classes will be detected as novel, because our novel class detection technique detects a class as novel only if it is found in the previously unused (unoccupied) feature spaces. However, if part of one or both of the new classes

occupies a new feature space, then those parts will be detected as novel. It is obvious that the data streams that do not have any fixed feature space (such as text stream) will have different feature spaces for different models in the ensemble, since different sets of features would likely be selected for different chunks. Besides, the feature space of test instances is also likely to be different from the feature space of the classification models.

8. References

1. C.C. Aggarwal, "On Classification and Segmentation of Massive Audio Data Streams," Knowledge and Information System, vol. 20, pp. 137-156, July 2009.
2. C.C. Aggarwal, J. Han, J. Wang, and P.S. Yu, "A Framework for On-Demand Classification of Evolving Data Streams," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 5, pp. 577-589, May 2006.
3. A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavalda`, "New Ensemble Methods for Evolving Data Streams," Proc. ACM SIGKDD 15th Int'l Conf. Knowledge Discovery and Data Mining, pp. 139-148, 2009.
4. S. Chen, H. Wang, S. Zhou, and P. Yu, "Stop Chasing Trends: Discovering High Order Models in Evolving Data," Proc. IEEE 24th Int'l Conf. Data Eng. (ICDE), pp. 923-932, 2008.
5. W. Fan, "Systematic Data Selection to Mine Concept-Drifting Data Streams," Proc. ACM SIGKDD 10th Int'l Conf. Knowledge Discovery and Data Mining, pp. 128-137, 2004.
6. J. Gao, W. Fan, and J. Han, "On Appropriate Assumptions to Mine Data Streams," Proc. IEEE Seventh Int'l Conf. Data Mining (ICDM), pp. 143-152, 2007.
7. S. Hashemi, Y. Yang, Z. Mirzamomen, and M. Kangavari, "Adapted One-versus-All Decision Trees for Data Stream Classification," IEEE Trans. Knowledge and Data Eng., vol. 21, no. 5, pp. 624-637, May 2009.
8. G. Hulten, L. Spencer, and P. Domingos, "Mining Time-Changing Data Streams," Proc. ACM SIGKDD Seventh Int'l Conf. Knowledge Discovery and Data Mining, pp. 97-106, 2001.
9. I. Katakis, G. Tsoumakas, and I. Vlahavas, "Dynamic Feature Space and Incremental Feature Selection for the Classification of Textual Data Streams," Proc. Int'l Workshop Knowledge Discovery from Data Streams (ECML/PKDD), pp. 102-116, 2006.
10. I. Katakis, G. Tsoumakas, and I. Vlahavas, "Tracking Recurring Contexts Using Ensemble Classifiers: An Application to Email Filtering," Knowledge and Information Systems, vol. 22, pp. 371-391.
11. J. Kolter and M. Maloof, "Using Additive Expert Ensembles to Cope with Concept Drift," Proc. 22nd Int'l Conf. Machine Learning (ICML), pp. 449-456, 2005.
12. D.D. Lewis, Y. Yang, T. Rose, and F. Li, "Rcv1: A New Benchmark Collection for Text Categorization Research," J. Machine Learning Research, vol. 5, pp. 361-397, 2004.
13. X. Li, P.S. Yu, B. Liu, and S.-K. Ng, "Positive Unlabeled Learning for Data Stream Classification," Proc. Ninth SIAM Int'l Conf. Data Mining (SDM), pp. 257-268, 2009.
14. M.M. Masud, Q. Chen, J. Gao, L. Khan, J. Han, and B.M. Thuraisingham, "Classification and Novel Class Detection of Data Streams in a Dynamic Feature Space," Proc. European Conf.Machine Learning and Knowledge Discovery in Databases (ECML PKDD), pp. 337-352, 2010.
15. M.M. Masud, Q. Chen, L. Khan, C. Aggarwal, J. Gao, J. Han, and B.M. Thuraisingham, "Addressing Concept-Evolution in Concept- Drifting Data Streams," Proc. IEEE Int'l Conf. Data Mining (ICDM)
16. M.M. Masud, J. Gao, L. Khan, J. Han, and B.M. Thuraisingham, "A Practical Approach to Classify Evolving Data Streams: Training with Limited Amount of Labeled Data," Proc. IEEE Eighth Int'l Conf. Data Mining (ICDM), pp. 929-934, 2008