A Semi-supervised Ensemble Approach for Mining Data Streams

Jing Liu [1,2], Guo-sheng Xu [1,2], Da Xiao [1,2], Li-ze Gu [1,2], Xin-xin Niu [1,2]
1.Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876,China
2.National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications,
Beijing 100876,China
Email: liujing81@sohu.com

Abstract—**There are many challenges in mining data streams, such as infinite length, evolving nature and lack of labeled instances. Accordingly, a semi-supervised ensemble approach for mining data streams is presented in this paper. Data streams are divided into data chunks to deal with the infinite length. An ensemble classification model E is trained with existing labeled data chunks and decision boundary is constructed using E for detecting novel classes. New labeled data chunks are used to update E while unlabeled ones are used to construct unsupervised models. Classes are predicted by a semi-supervised model Ex which is consist of E and unsupervised models in a maximization consensus manner, so better performance can be achieved by using the constraints from unsupervised models with limited labeled instances. Experiments with different datasets demonstrate that our method outperforms conventional methods in mining data streams.**

Index Terms—**data stream mining, semi-supervised learning, novel class, concept drifting**

I. Introduction

Tremendous amount of data are generated by many information systems at unprecedented rates, such as network security events and logs, social text streams, credit card transactional flows, surveillance video streams and so on. Mining such data streams are usually challenged by some properties: infinite length, evolving nature and lack of labeled data. The evolving nature includes concept drift, novel classes appearing, outdated classes disappearing and recurring classes. There are mainly two categories of solution for mining data streams, which are single model incremental learning algorithms and ensemble learning ones. Single model algorithms maintain a model which uses new data for update to adapt to the evolving nature of data streams. Ensemble learning methods train some base models using some labeled data chunks firstly, then an ensemble model combining all base models can be obtained. New labeled data chunks are used to train new base models and old models who behave poorly in identifying new data chunks will be replaced. Ensemble learning methods are more popular than single model ones for their simpler implementation and higher efficiency.

Many researches have been carried out to deal with the evolving nature of data streams. A machine learning algorithm is proposed in [1] which deals with the changing concepts by mining decision trees from data streams by time windows. Ref.[2] proposes a general framework using weighted ensemble classifiers. But it's difficult to identify historical examples which share identical distributions to the test data without prior knowledge. Ref.[3] assumes that data chunks which are temporally are similar with each other, so the most up-to-date chunks are used to predict the yet-to-arrive ones. Ref.[4] suggests an AddExp algorithm which updates the weights of base models steadily and adds a new one when the ensemble misclassifies an instance. A framework for studying concept-drift data streams is proposed in [5] as well as two new variants of Bagging. These data stream classification techniques address only the infinite length and concept-drift problems and they cannot recognize novel class instances until it is manually identified. Some other researches consider the concept-evolution problem as well. Ref.[6] proposed an efficient technique that can automatically detect novels classes by quantifying cohesion among unlabeled test instances and separation of the test instances from training instances using discrete Gini Coefficient. Ref.[7] studies how to detect novel classes in concept-drifting data streams under time constraints, including a time constraint to classify a test instance and a time delay of manual labeling an instance. Ref.[8] introduce a novelty detection technique named OLINDDA that applied to intrusion detection in computer network. Ref.[9] propose an approach for incremental learning by building a decision tree model from training dataset, the tree continuously updates so that it represents the most recent concept in data stream. Ref.[10] considers the detection of recurring classes which reappear after long disappearance from the stream. Though these algorithms can detect novel classes, they require adequate labeled instances in data streams, which is impractical since manual labeling of data is both costly and time-consuming. All above algorithms suffer from the lack of labeled instances.

Many researches on how to mining data streams combing labeled and unlabeled data have also been carried out. Ref.[11] builds a micro-clusters based classification model using semi-supervised technique from a training set having only a small amount of labeled instances, an ensemble of these models is used to classify the unlabeled instances with kNN algorithm. But the labeled data should cover all classes in this algorithm, which is a difficult issue. Ref.[12] proposes a RK-TS^{3}VM model intending to leverage labeled and unlabeled samples to build prediction models, which is a relational k-means based transfer semi-supervised SVM learning framework. Unfortunately, It cannot deal with the classes not appeared in the labeled instances either. Ref.[13] proposes an ensemble model combining both classifiers and clusters together to improve performance. It uses a label propagation method to infer each cluster's class label and a weighting schema to weight all base models like a special PageRank. But it assumes that the classes in labeled data chunks are identical with the classes in unlabeled ones, which cannot be satisfied strictly in real environment. Ref.[14] presents a semi-supervised learning based ensemble classifier utilizing both a small fraction of labeled instances and a great deal of unlabeled ones. But the classes in the data stream are assumed to be fixed. In addition, some methods assume that labeled instances can be obtained in almost every data chunk, but it is difficult to label even a part of instances in every data chunk in practical applications. So new methods which are more feasible in real environment should be studied.

Accordingly, a semi-supervised ensemble approach considering all the challenges mentioned above is presented in this paper. Firstly, the data streams are divided into equal-sized data chunks. Then, some base models are trained using existing labeled data chunks and then combine them into an ensemble model *E*. A decision boundary using *E* is obtained including all classes that have ever been appeared. If the new coming data chunk is labeled, it will be used to update *E* and the decision boundary. Contrarily, if it is unlabeled, we use *E* to justify the classes in the data chunk as well as the detection of novel classes. Then unsupervised models can be obtained from unlabeled instances and they can provide useful constraints for *E* to refine the final hypothesis.

The rest of the paper is organized as follows: Section II describes the data stream model; Section III formulates the proposed approach in detail; section IV reports the experimental results; We concludes the paper in section V.

II. Overview of the Model

The data stream model is given for the convenience of narration. A data stream *D* is an infinite sequence of instances (x_i, y_i), each instance x_i is a d-dimensional vector with a class label $y_i \in Y = \{y_1, \dots, y_c\}$. To deal with the infinite nature of data stream, we split *D* into equal-sized data chunks $D=\{D_1, D_2, \dots, D_i, \dots\}$ with chunk size *n*. Most of the data chunks are leaved unlabeled (or partially labeled) as in Fig.1.

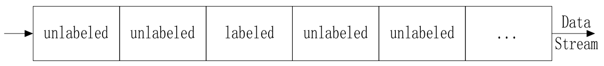

Figure1. Data Stream with limited labeled instances

$\{\lambda_1, \lambda_2, \dots, \lambda_c\}$ are classifiers that have been trained with the labeled data chunks, c express the total number of classes seen so far in *D* and there is one classifier λ_i for each class. The ensemble model $E = \{\lambda_1, \lambda_2, \dots, \lambda_c\}$ constitutes the complete classification model. Let $Y = \{y_1, y_2, \dots, y_c\}$ be the set of all class labels in all the labeled data chunks in the stream. In other words, for every $y_i \in Y$, there is at least one labeled chunk used for training class y_i. Likewise, a class *y* is a novel class if it has never appeared in *D* so far, so it can be concluded that $y \notin Y$.

If the coming new data chunk D_i is labeled by human experts, it is used to train new models for each class contained in D_i and the corresponding $\lambda_i \in E$ are updated. When a novel class appears in D_i, it will be added to the list of classes and a corresponding λ and *y* will be added to *E* and *Y* separately. On the contrary, if the coming data chunk D_i is unlabeled or partly labeled, the unlabeled instances in the chunk are roughly classified by *E* firstly. Letting $E_u = \{\lambda^1, \lambda^2, \dots, \lambda^\alpha\}$ are a base unsupervised models from D_i. Though these unsupervised models in E_u cannot give class label of instances in D_i directly, it can provide useful constraints on the label prediction for each instance. A joint semi-supervised ensemble model $E_x = E \cup E_u$ is formed to classify the instances in the unlabeled chunk D_i precisely in a maximization consensus manner. Additionally, it should be noted that classification of incoming unlabeled data chunk is a continuous process in data streams, so it cannot stopped for training and updating of the classifiers in *E*. Therefore, novel classes are labeled as ϕ temporarily until the corresponding labeled data chunk arrived.

III. Methodology

A. Training and Updating E with Limited Labeled Data Chunks

It is memory-inefficient to store all the raw labeled data for training and time-inefficient to use them to classify unlabeled data chunks. We reduce both the time and memory requirement by saving sketch information of every class, and discarding the raw data.

A labeled training data chunk D_i is split into *r* partitions $\{s_1, s_2, \dots, s_r\}$ based on class labels firstly, where *r* is the total number of classes contained in the chunk. *K* clusters are built with each partition using K-means clustering. For each cluster, sketch information is extracted including the centroid μ, the radius *R*, and the number of instances *N* of the cluster, which is called micro-cluster. The set of micro-clusters constitute a model M^i for each class *i*. An ensemble model E^i contains *L* such models for each class, denoted by $E^i = \{M_1^i, M_2^i, \dots, M_L^i\}$. So the collection of ensembles is $E = \{E^1, E^2, \dots, E^c\}$, where *c* is the total number of classes that have been appeared in *D* so far. Each micro-cluster mentioned above corresponds to a hypersphere in the d-dimensional feature space, so the decision boundary of class *i* is the union of all such hyperspheres in corresponding M^i. The decision boundaries of all classes constitute the decision boundary of E. If the distance of an unlabeled instance to the centroid of a micro-cluster is less than or equal to the radius of the micro-cluster, it is considered to have the corresponding class label of the micro-cluster. When a new coming data chunk D_n is labeled, new models of classes contained in it are trained, and the number of models for each corresponding class becomes L+1. L models are chosen based on the prediction accuracy on the latest labeled data chunk to update the corresponding models, which ensures that a constant amount of memory is required to store L models in each ensemble.

The construction of the ensemble model E solves the problem of infinite length, as well as the limited memory. The concept-drift problem is addressed through the update of E. But E can only classify instances of classes with which they have been trained. All instances belonging to a new class will be misclassified until a new base model is trained with the labeled instances of that class. So a novel class detection method should be incorporated to identify the emergence of novel classes.

B. The Detection of Novel Classes

The central concept of novel class detection is that each class must have an important property: the data points belonging to the same class should be close to each other and should be far apart from the data points belonging to other classes. There are two main stages in the classification process, named outlier detection and novel class detection. When an unlabeled data chunk arrives, the decision boundary of E is used to classify the instances in the chunk. If an instance falls outside the decision boundary of E, it is considered as an outlier and will be temporarily stored in a buffer for further inspection.

Several computations on the outliers in the buffer are performed to detect the arrival of a new class. For every outlier x, we define $\lambda_c(x)$ to be the set of N-nearest neighbors belonging to class c where N is a predefined parameter. Letting $b_c(x)$ be the average distance from x to the instances in $\lambda_c(x)$ and $a(x)$ be the average distance from x to the set of N outliers that are closest to x. Using this neighborhood information, we compute a metric named N-neighborhood silhouette coefficient (*N-NSC*) as

$$N-NSC(x)=\frac{b_{min}(x)-a(x)}{\max(b_{min}(x),a(x))}\ . \qquad (1)$$

where $b_{min}(x)$ is the minimum among all existing $b_c(x)$.A positive value of *N-NSC* demonstrate that x is closer to its own class and farther away from other existing classes. A new class is declared if there are at least N' outliers whose *N-NSC* value is positive, where N' is a predefined value greater than N.

C. The Consensus Maximization Method with Ex

It is often the case that we may have only a small portion of labeled data chunks to train ensemble model E due to the high cost of manual labeling by experts. Meanwhile, a large number of unlabeled data chunks in data streams are available to build unsupervised models. Although these models cannot directly predict class label for unlabeled instances, they can provide useful constraints leading to better prediction performance.

Without loss of generality, considering an unlabeled data chunk containing r classes and we have a unsupervised models $\lambda^1,\lambda^2,\dots,\lambda^a$ in all. Then our aim is to combine ensemble classifier E and these unsupervised models to construct a semi-supervised ensemble model E_x which has better prediction performance than E. Then for every instance x in the new coming unlabeled data chunk, E_x will predict the class label y^* which satisfies following equation:

$$y^*=argmax_{y\in Y}P(y|x,E_x). \qquad (2)$$

The posterior probability $P(y|x,E_x)$ is the weighted sum of E and all unsupervised models as shown in the following equation:

$$P(y|x,E_x)=w_0P(y|x,E)+\sum_{i=1}^{a}w_iP(y|x,\lambda^i). \quad (3)$$

In the above equation, ensemble E can give x a label directly while λ^i can only assign x a cluster id g_k^i without any class label information. So a variable $P(y|g_k^i)$ is introduced to bridge the cluster id and class label. Assume that a data chunk is partitioned into r clusters by each base model λ^i, and there are a total of $v=r(1+a)$ clusters, where the first r clusters are generated by E and the remaining ra clusters are generated by unsupervised models. Then the posterior probability $P(y|x,\lambda^i)$ can be expressed as follows:

$$P(y|x,\lambda^i)=\sum_{k=1}^{r}P(y|g_k^i)P(g_k^i|x,\lambda^i). \qquad (4)$$

(3) can be revised to

$$P(y|x,E_x)= \\ w_0P(y|x,E)+\sum_{i=1}^{a}\sum_{k=1}^{r}w_iP(y|g_k^i)P(g_k^i|x,\lambda^i)\ . \qquad (5)$$

where the first term represents the classify result of E and the second term represents the constraints of all unsupervised models. It is assumed that these models provide complementary expertise, so their maximization consensus should represents a better solution than single E.

Some notations are needed to solve the maximization consensus problem. Matrix $A_{n\times v}$ denotes the clustering result of all base models, including E and unsupervised models. Each entry a_{ij} is the normalized probability of x_i being assigned to group j by a model. $U_{n\times r}$ is a matrix containing the confidence information associated with the final label prediction result of E_x, for example, $u_{iz}=1$ indicating that x_i is assigned to class z by E_x. $Q_{v\times r}$ indicates the probability of a cluster j is assigned to a class z. In addition, the first r clusters are different from others in that they are obtained from a classify ensemble E, which containing the initial class label information needed by the proposed approach. So it is specially denoted by a matrix $Y_{v\times r}$. More generally, if some of the instances(m for example) in D_i are labeled by experts, then a matrix $F_{m\times r}$ can be obtained. Then we can formulate the maximization consensus problem as following:

$$\min\psi(Q,U)=\sum_{i=1}^{n}\sum_{j=1}^{v}a_{ij}||\boldsymbol{U_i}-\boldsymbol{Q_j}||^2+\alpha\sum_{j=1}^{v}||\boldsymbol{Q_j}-$$

$$\boldsymbol{Y_j}||^2 + \beta \textstyle\sum_{i=1}^{n} ||\boldsymbol{U_i} - \boldsymbol{F_i}||^2. \tag{6}$$

where α is a parameter controlling the influence of classification ensemble *E* and β reflects the influence of the labeled instances. The first term implies that the propagation of label information among all clusters is according to the internal structure. The second term ensure that the maximization consensus label prediction of E_x should not deviate much from the initial estimation result provided by *E*. The third term penalizes the deviation of U_i from the labeled instances denoted by F_i. Finally, $\alpha = 0$ if the class predicted by *E* is ϕ(novel class) and $\beta = 0$ if the corresponding instance is unlabeled.

It can be seen from (6) that the value of $\psi(Q, U)$ is decided by two parameters *Q* and *U*. We use a iteration method to solve the problem. If we fix the value of *U*, $\psi(Q, U)$ will be an objective function with respect to *Q*. Then the value of *Q* that can give the minimum $\psi(Q, U)$ is as follows:

$$\boldsymbol{Q_j} = \frac{\sum_{i=1}^{n} a_{ij}\boldsymbol{U_i} + \alpha Y_j}{\sum_{i=1}^{n} a_{ij} + \alpha}. \tag{7}$$

Likewise, when *Q* is fixed, the *U* value corresponding to minimum $\psi(Q, U)$ is:

$$\boldsymbol{U_i} = \frac{\sum_{j=1}^{v} a_{ij}\boldsymbol{Q_j} + \beta \boldsymbol{F_i}}{\sum_{j=1}^{v} a_{ij} + \beta}. \tag{8}$$

The initial value of *U* is given by the classification ensemble model *E*, and the value of *Q* can be obtained using (7), then a new value of *U* can be acquired with (8). This iteration process continues until (*Q,U*) converges to a stationary point of the $\psi(Q, U)$ minimum problem. Finally, the maximum value in the vector U_i is the prediction label of the instance *i* in the data chunk.

As mentioned above, we summarize the procedure of building and updating ensemble model E_x using classification ensemble model *E* and unsupervised models in the following algorithm 1. When a new arrived data chunk is labeled, it is used to construct a new base classification model and update *E*. Otherwise, if it is unlabeled or partly labeled, we first use *E* to classify the unlabeled instances roughly and decide the number of classes contained in it. Then unsupervised models are used to provide useful constraints for precisely decision that satisfies maximization consensus. The decision result is acquired in an iteration manner.

```
Algorithm 1: The Semi-supervised Ensemble Approach for Mining Data Streams
Input: A data stream $D$; Data chunk size $n$; Unsupervised models $(\lambda^1, \lambda^2, \ldots, \lambda^a)$;
Output: The classification result of the instances in $D$.
1: Split $D$ into equal-sized data chunks
2: Build classification ensemble model $E = \{\lambda_1, \lambda_2, \ldots, \lambda_c\}$ with the initial labeled data chunks
3: while $D$ not empty
4:   if the new arriving chunk $D_i$ is labeled, then update $E$ as in section3.1
5:   if $D_i$ is unlabeled
6:       Roughly classify $D_i$ and decide the number of classes in $D_i$, denoted by $r$ as in section 3.2
7:       Construct unsupervised models using $r$
8:       Construct extend ensemble model $E_x$ using $E$ and unsupervised models
9:       Solve the optimized classification result using (7) and (8) in section 3.3
10:  end if
11: end while
```

IV. Simulation Results and Analysis

In this section, experimental studies are reported to verify the performance of the proposed method.

A. *Data Sets and Experimental Setup*

We apply the proposed approach on synthetic datasets as well as two real datasets, kddcup99 intrusion detection dataset and forest cover type dataset from UCI repository. The construction of synthetic datasets can be referred in [6], which simulate the concept-drifting and novel class appearance through the shifting of parameters. Each synthetic dataset has 40 real valued attributes and 100,000 data points. A synthetic dataset having *X* classes is denoted as *SynCX*. We use 10% of the kddcup99 dataset containing about 490,000 instances, 23 classes (22 attacks and 1 normal). Each record contains 42 attributes and 34 numeric attributes are used in experiments. Forest cover type dataset contains more than 580,000 instances. Each record contains 54 attributes and 10 quantitative variables are used in this experiment. Different classes appear and disappear frequently in both synthetic and real datasets, making classes in different data chunks variable. We report the average results of 10 different randomly generated sequences of each dataset.

The values of relative parameters are set as follows: the size of data chunk *n*=1000; the ensemble size used to train base classifier *L*=3; the number of unsupervised models *r*=3; the value used to declare a novel class *N*=50; the values reflect the influence of *E* and labeled instances are $\alpha = 2, \beta = 6$ seperately; the proportion of labeled data chunks is 20% ; 5% labeled instances are randomly distributed in other data chunks. The algorithm is implemented in Weka platform based on Java. The experiments run on a computer with 3.2GHz dual processor CPU and 4GB memory. To comparing with our method, we select three competing algorithms: WCE proposed in [2], SCANR proposed in [10] and ECU proposed in [13]. Our proposed method is denoted by SSEA(Semi-Supervised Ensemble Approach) for simplification in experiments.

B. *Performance Analysis*

Table1 lists the results of all methods on different datasets with 20% labeled data chunks. The result of each dataset for a particular algorithm is acquired by averaging the result of 10 independent experiments.

TABLE I
SUMMARY RESULT

	WCE	SCANR	ECU	SSEA
SynC20	92.6%	95.8%	94.3%	98.7%
SynC40	86.2%	93.7%	89.2%	98.1%
Kddcup99	63.5%	82.9%	79.4%	89.3%
Forest cover	59.7%	70.2%	67.1%	78.8%

It can be seen from the above table that the proposed approach SSEA always has the best performance in all experiments, while WCE performs worst, followed by ECU. The main reason for the error of WCE is that it fails to detect novel class instances, and ECU has the same cause. ECU improves the performance by combining unlabeled clustering models together with classifying ensemble, which demonstrate that unsupervised models can provide useful constraints. Since SCANR can adapt the evolving nature of the data stream, so it can improve the prediction accuracy through correctly identifying most of the novel class instances and recurring class instances. The reason why SSEA is prior to SCANR is that SSEA uses the unlabeled data chunks reasonably while SCANR leave these chunks unused.

Figure 1 shows the prediction accuracy comparisons among these algorithms in a chunk-by-chunk manner on a section of kddcup99 and forest cover type dataset. The results also validate that SSEA performs best compared to other algorithms with limited amount of labeled data chunk.

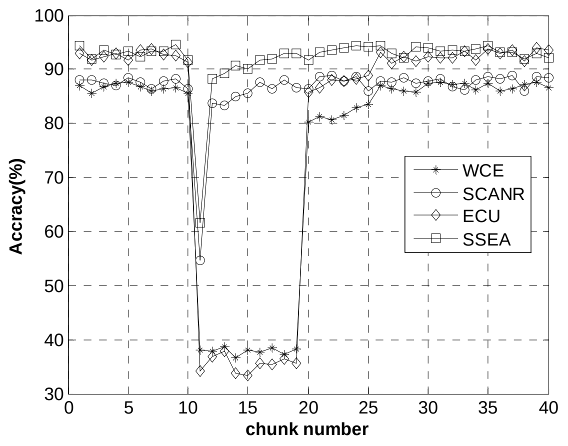

Figure2. Accuracy comparison chunk-by-chunk on kddcup99

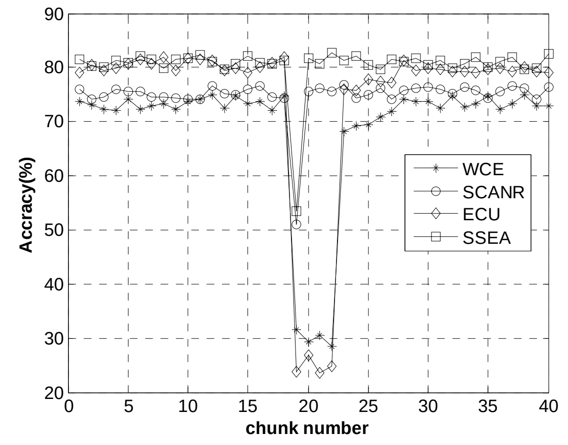

Figure3. Accuracy comparison chunk-by-chunk Forest cover

In addition, it can be seen in Fig.2 and Fig.3 that when concept-evolution (chunk 11 in kddcup99, chunk 18 in forest cover) occurs in some data chunks, the current ensemble model cannot adapt to the new data chunk, so all algorithms will suffer from performance degradation in different extent. SSEA and SCANR can detect novel classes and recurring classes, which can ease the degradation at a certain extent and quickly recover from the bottom point. WCE and ECU cannot detect novel classes, so they suffer more from concept-evolution and cannot recover until a labeled training chunk arrives (chunk 20 in kddcup99, chunk 21 in forest cover). Especially, ECU degrades most in the case of concept-evolution because ECU mistakenly classify all novel class instances until the new training data arrives, so the unsupervised models may provide harmful information in this period. The performance of all algorithms will recover when the ensemble model is modified using new training data chunks. When the nature of the data stream is smooth in a period, SSEA and ECU will perform better for the useful constraint provided by unsupervised models.

C. *Parameters' Influence*

For further research on the proposed algorithm SSEA, we test the sensitivity of the algorithm on kddcup99 dataset by varying some parameters including the proportion of labeled data chunks p, the chunk size n, the initial ensemble size L , the number of unsupervised models r, the influence control factors α and β. These parameters should have similar effects on experiments on other datasets.

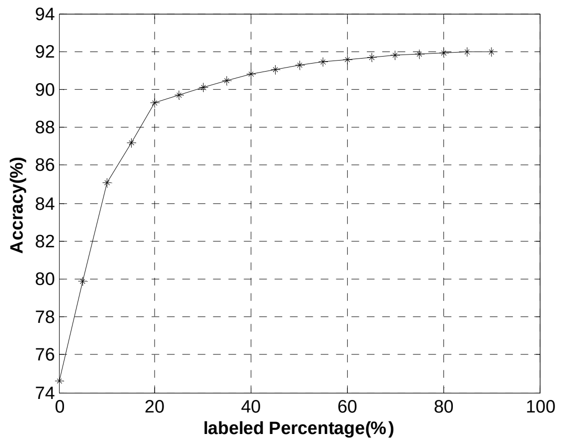

Figure4. Prediction accuracy VS labeled percentage

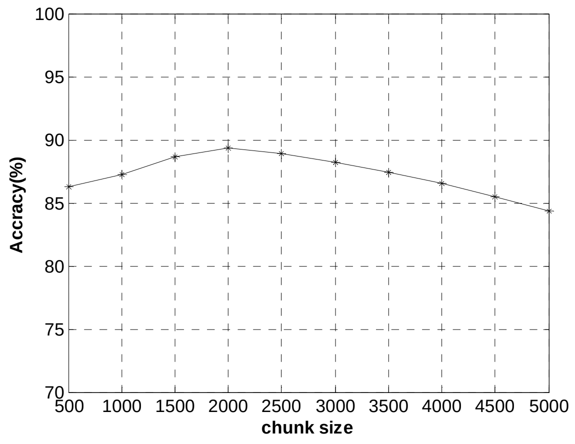

Figure5. Prediction accuracy VS chunk size

Fig.4 indicates that the prediction accuracy increases with p but the increasing rate keeps decreasing all along, which is quite slow after 20%. So the proposed SSEA can acquire quite good performance with limited labeled instances. In real applications where labeled instances are difficult to obtain, SSEA can achieve better performance with limited labeled instances. Fig.5 shows the effect of n. It can be seen that the prediction accuracy increases along with n upto about 2000 and then decrease. The reason is that large labeled data chunk can provide more training data for E, which can promote the quality of the model. But if the chunk becomes too large, longer period of time will be needed to update the ensemble model and more instances will be classified with higher error rate.

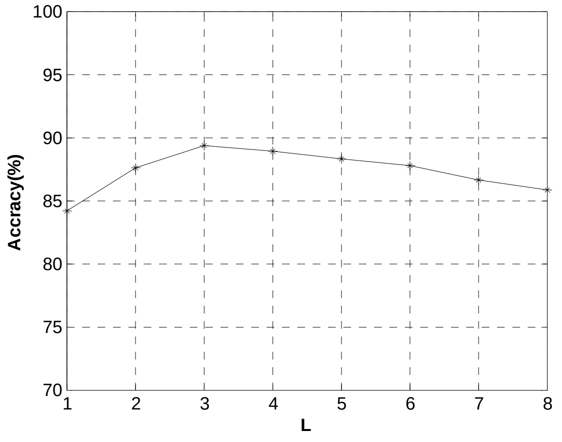

Figure6. Effect of the number of supervised models L

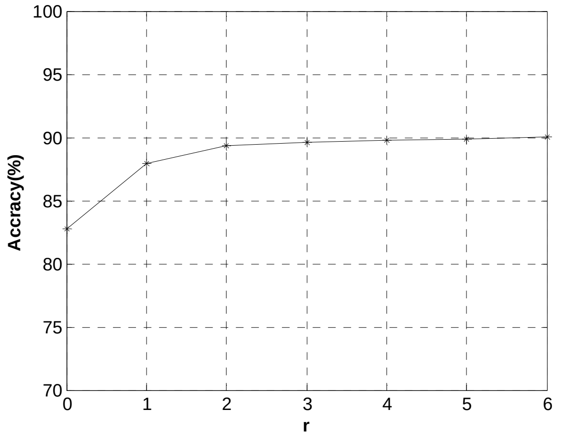

Figure7. Effect of the number of unsupervised models r

The effect of L is shown in Fig.6. We can see that the prediction accuracy increases along with L slowly and begin to decrease at 3. The incensement owe to the reduction of error variance. The reason for the performance degradation is that too large L making the novel class detection condition rigorous and more novel class instances will be classified mistakenly. Fig.7 shows that more unsupervised models help increasing prediction accuracy slowly. It can be inferred that a larger r means the algorithm using the unlabeled data chunks more sufficiently. Since larger r brings more computation and the increment is diminished gradually, we don't select too large r in experiments.

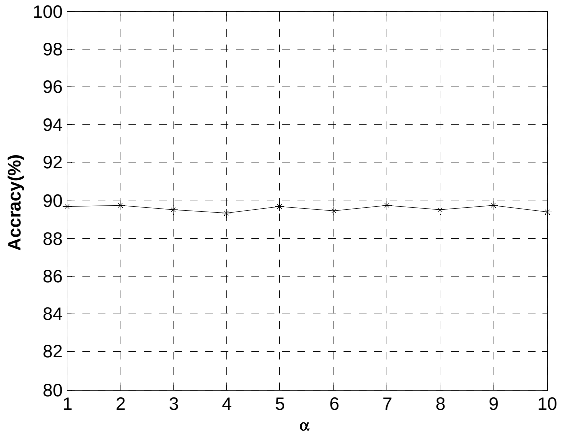

Figure8. Sensitive analysis of α

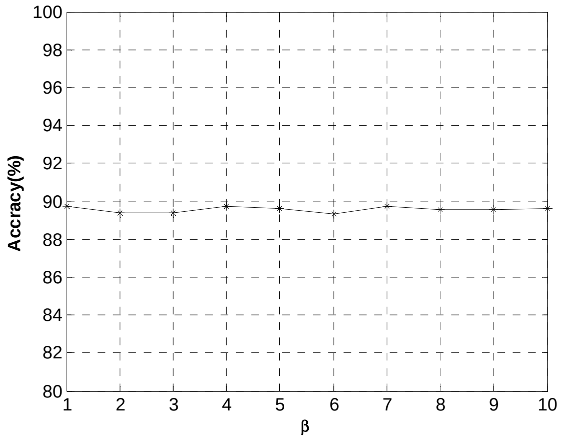

Figure9. Sensitive analysis of β

It can be seen from Fig.8 and Fig.9 that SSEA is not sensitive to α and β. As mentioned in section 3.3, α represent the confidence of prediction result provided by E and β reflect the prices paid for deviating from the labeled instances. Since the labels provided by E may be incorrect while the labeled instances always true. So we take a smaller α and a larger β in the experiments.

V. CONCLUSION

In this paper, we propose a semi-supervised ensemble approach which can improve the data stream classification performance using limited labeled instances. The essential goal of the approach is to boost the performance by making full use of labeled and unlabeled instances. The goal is achieved by maximizing the consensus between the supervised ensemble classification model E from labeled instances and unsupervised models from unlabeled ones by propagating label information among them iteratively. Additionally, the approach can use initial labeled instances to guide the propagation. The influence of different components can be tuned by parameters. Experimental results with both synthetic and real datasets indicate the benefits of the proposed method over traditional competing algorithms, and the influences of different parameters are analyzed to find the optimal values. In further research, the features of data streams will be incorporated to deal with the feature-evolution problem which often emerges in real scenario as network intrusion detection.

ACKNOWLEDGEMENTS

This research has been partially supported by National Natural Science Foundation of China(No.61202082)and the Fundamental Research Funds for the Central Universities (No.BUPT2012RC0218). The views of this paper should not be interpreted as representing any funding agencies.

REFERENCES

[1] G. Hulten, L. Spencer, and P. Domingos, “Mining time-changing data streams,” in Proc. SIGKDD, 2001, pp. 97–106.

[2] H. Wang, W. Fan, P. S. Yu, and J. Han, “Mining concept-drifting data streams using ensemble classifiers,” in Proc. KDD ’03 , 2003, pp. 226–235.

[3] J. Gao, W. Fan, and J. Han., “On appropriate assumptions to mine data streams.” In Proc. ICDM, 2007, pp. 143–152.

[4] J. Kolter and M. Maloof., “Using additive expert ensembles to cope with concept drift.” in Proc. ICML , Aug 2005, pp.449–456.

[5] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavald,“New ensemble methods for evolving data streams,” in Proc. SIGKDD , 2009, pp. 139–148.

[6] M. M. Masud, Q. Chen, L. Khan, C. C. Aggarwal, J. Gao, J. Han, and B. M. Thuraisingham,“Addressing concept evolution in concept-drifting data streams,” in Proc. ICDM, 2010, pp.929–934.

[7] M. M. Masud, J. Gao, L. Khan, J. Han, and B. M. Thuraisingham, “Classification and novel class detection in concept-drifting data streams under time constraints,” IEEE TKDE, vol.23, no.1, pp. 859–874, 2011.

[8] E. J. Spinosa, A. P. deLeonF. deCarvalho, and J. Gama. “Cluster-based novel concept detection in data streams applied to intrusion detection in computer networks,” In Proc. ACM SAC, pages 976–980, 2008.

[9] D. Farid, and C. M. Rahman.” Novel Class Detection in Concept-Drifting Data Stream Mining Employing Decision Tree,” in Proc. ICECE, 2012,pp.630–633.

[10] M.M.Masud,T.M.Al-Khateeb,L.Khan,C.C.Aggarwal,J.Gao,J.Han,andB.M.Thuraisingham,“Detecting recurring and novel classes in concept-drifting data streams,” in Proc.ICDM’11, Dec.2011, pp. 1176–1181.

[11] M. Mohammad, J. Gao, L. Khan, and J. Han. A practical approach to classify evolving data streams: training with limited amount of labeled data. In Proc. of IEEE ICDM, 2008.

[12] P. Zhang, X. Zhu, and L. Guo. “Mining Data Streams with Labeled and Unlabeled Training Examples,”. In Proc. of ACM KDD, 2008, pp.627-636

[13] P. Zhang, X. Zhu, J. Tan, and L. Guo. “Classifier and Cluster Ensembles for Mining Concept Drifting Data Streams,”. In Proc. of IEEE ICDM, 2010, pp.627-636

[14] W. H. Xu, Z. Qin, and Y. Chang.” Semi-Supervised Learning Based Ensemble Classifier for Stream Data, ” PR and AI, vol.25, no.2, ,2012, pp. 292–299.

Jing Liu, is a PhD candidate in the Information Security Center, Beijing University of Posts and Telecommunications. His research interests include network security, data analysis and cloud computing.

Guosheng Xu is an assistant professor of Computer Science and Technology at Beijing University of Post and Telecommunications, China. He received his PhD degree in Cryptography from Beijing University of Posts and Telecommunications in 2008. His main research interest are modern cryptography and software design.

Da Xiao, is an assistant professor of Computer Science and Technology at Beijing University of Post and Telecommunications, China. He received his BS and Ph.D. degrees in Computer Science and technology from Tsinghua University, China in 2003 and 2009, respectively. His research interests include cloud storage, storage security, distributed storage and file systems, disaster backup and recovery, etc.

Lize Gu, is an associate professor of Computer Science and Technology at Beijing University of Post and Telecommunications, China. His main research interest are modern cryptography, E-commerce and network security.

Xinxin Niu, is a professor of Computer Science and Technology at Beijing University of Post and Telecommunications, China. Her research interests include information security, digital content security and software defined radio, etc.