Active Learning From Stream Data Using Cost-Oriented Weight Classifier Ensemble

Wenjia Xie* wenjiaxi@usc.edu
Di Yang diy@usc.edu
Xin Zhang zhan413@usc.edu

1. PROPOSAL

1.1 Introduction

Recent developments in storage and networking technology have made it possible for broad areas of applications on stream data for rapid decision making [25]. In the domain of classification, it is essential to provide a set of labeled training examples for generating predictive models. However, labeling all data is considered expensive and impractical, and the priori and posterior probability of a class and the class conditional probability may constantly drift across the scream [23][27]. A common practice to solve the problem is to use active learning techniques to selectively label a number of instances from which an ideal model is derived to predict future instances [24]. Some recent works unify the two aspects (effort and gain) of active learning by studying active learning in the cost-oriented framework [11] [18]. However, these works only focus on a ′pool-based′ static data set rather than data streams in our research. Other recent works mainly focus on another performance measure, accuracy. Though these works are based on active learning from data streams, they havenaŕt taken cost of errors into consideration, which may not be appropriate in many real-world applications as different kind of errors may lead to quite different consequences. So our object is to apply cost-oriented classifier to data streams and remain a relative high accuracy. An active learner generally begins with a very small percent of randomly labeled instances, chooses some additional instances to be labeled, learns from the results of those additional examples to build classifier ensemble, and then chooses which instances to label next through the newly built classifier. Through the active learning procedures we can minimize the total cost by labeling only a few ′important′ instances from the data stream of a recent period of time.

1.2 Innovations

As ensemble learning and active learning are both will developed areas, thus many related works on each topic published these years[17]. However, the paper combining of both of them are very rare and most of them failed to apply their methods to data stream until 2010[27]. Admittedly the approaching of [27] was very innovative but it is obviously that they was not able to include lost in their evaluation. As will be further discussed in part 5, focusing only on accuracy will not satisfy most cases in real life so we would like to propose a new frame that makes the lose to be minimized.

1.3 Difficulties and Challenges

Giving the unique properties of stream data as described above, several problems that current solution, usually focusing on static data processing, are unable to deal with emerge. As the change of data pool which data stream provides, the $p(x|c_i)$ and $p(c_i)$ (where x indicates a sample and c_i indicates the class where $i \in 1, 2$.Thus $p(x|c_i)$ indicates the possibility of $x \in i$)also alerting with the observed data sets leading to the mutation of posterior probability as the equation(1)below shows:

$$\arg \max_{i \in (1,2)} p(c_i|x) = \arg \max_{i \in (1,2)} \frac{p(x|c_i)p(c_i)}{p(x)} \quad (1)$$

Thus, a static model will not fit data string. So the model of classification should have the ability to transform themselves in accordance to the recent data. The transformation of model must satisfy two important issues data stream bringing to us.

First of all, with the increasing of time, the optimal of our classification problem will be different from the previous one. In another word, the distribution of data itself will change with time. To solve this problem we need new criterion to select a new set of labeled samples buy active learning and to train a brand new classifier at every epoch. However, doing this will cause the second problem we want to discuss that it is impossible for our resources to store the volume of data-set and classifiers increasing with time. As a result, we must find a way to combine all previous experience, namely the information we already gained, with newly discovery with very little redundancy, known as ensemble learning.

Although ensemble learning and weighted classification training is well developed, in this paper, while both training every classifier and ensembling them together, we will take the cost into consideration. There is still no mutual solution for ensemble classifier trained from every trunk of stream data cost-orientedly. This is also the most vital question we will work on this paper. Beside these, the definition of lost under each misclassification condition is hard to justify as well. Whereas, in this paper we only provide a general way

Table 1: Milestones and Schedule

Index	Time Period	Milestone
1	Oct.2 to Oct.3	Get familiar with Weka platform.
2	Oct.4 to Oct.9	Employ a hyperplane-based data stream generator to get data stream.
3	Oct.10 to Oct.17	Implement the data labeling in one chunk.
4	Oct.18 to Oct.25	Implement the active learning in one chunk.
5	Oct.26 to Nov.2	Explore the classifier ensemble method based on loss function.
6	Nov.3 to Nov.17	Evaluate the ensemble method with experiment.
7	Nov.18 to Dec.3	Prepare for the presentation.
8	Nov.4 to Dec.10	Implement the final report.

of general problem meaning that to each specific case the user who use this algorithm must carefully choose the lost value for each class.

1.4 Motivation

Fraud detection and fraud prevention have attracted people′s attention in recent years, especially when you consider credit card transactions from payment card users[26]. The reason for this increase in research activity can be attributed to the huge annual financial losses of banks by compensating for clients. We can assume that the cost of making a phone call for transaction confirmation is 10 dollars (including labor cost, equipment cost and maintenance cost, etc.). However, if we cannot find an unusual transaction, banks probably undertake to compensate for thousand dollars which is a huge loss. Therefore, a practical method for dealing with fraud relies on an online transaction detection center that monitors the incoming transaction data flow of card users to identify suspicious data based on abnormal financial records. Here, hourly transaction data flow constitutes a data stream. Lets say the number of financial records arrives at an average rate of 100,000 per hour, out of which the bank staff can only investigate 500. Accordingly, we apply active learning to the cost-oriented framework to decide which part of records should be investigated to improve the existing classifier model and make sure future loss as less as possible.

1.5 Improvement

Regarding to many real world applications, different kinds of errors such as false positive and false negative in binary classification can cost very differently so that all phone calls wont subject to the same cost. Thus, accuracy (as a performance measure) may not be appropriate. Also, the performance of a learning program can be evaluated by many different measures: accuracy, recall, computation time, and so on. However, as long as they can be converted to the cost, they can be included in the total cost of the existing learning system. Meanwhile, as active learning is converted to the minimization problem, many optimization techniques (such as gradient descent) can be used. Finally, and most importantly, the ultimate goal of most real-world application is to minimize the total cost (or maximize the total profit)[22]. In conclusion, studying active learning in the cost-oriented framework can bring many unique and important advantages.

1.6 Milestones and Schedule

See Table 1.

2. SURVEY

2.1 Origin of Active Learning

With the development of technology and increasing specific data applications, there are many situations in which unlabeled data is abundant but manually labeling is very expensive [21]. For example, if we want to implement a speech recognizer, we can record an original speech sample easily. However, we need to collect huge numbers of speech samples and label them to build an acoustic model which is not economical and very tedious for trained linguists. Therefore, we proposed to employ a learning algorithm that is able to actively query the expert for labeling. This type of learning is called active learning.

The most widely-used type of active learning is called pool-based active learning. A pool of large number of unlabeled example is given, and the learner may begin with a small number of instances in the labeled training set, request labels for one or more carefully selected instances, learn from the query results, and then leverage its new knowledge to choose which instances to query next [27]. The new labeled instance is simply added to the labeled set when a new query has been made, and the learner proceeds from there in a standard supervised way.

The reason motivates pool-based active learning is that large collections of unlabeled data can be gathered statically for many real-world learning problems. In fact, many works have been published in recent years on pool-based active learning, for example, Tong and Koller applied this learning algorithm in text classification, Settles and Craven use active learning to extract information, Zhang and Chen studied image classification and retrieval through active learning, etc.

2.2 Introducing active learning to data stream

For some real world learning instances, we cannot gather the whole collection of unlabeled data all at once. For example, considering an online credit card detection center that monitors the incoming traffic flow of payment records based on time, address and amount. Assume in the hourly traffic flow, which composes the data stream, the number of payment records arrive at an average rate of 100000 per hour, but only 5% of them can be investigated by technique staff to determine whether they are normal or not. The question is which 5% we should label as important instances so that we can identify the credit records as accurate as possible, or minimize the total cost. Therefore, an alternative to pool-based sampling is selective sampling [2].

When we refer to selective sampling, the unlabeled instance can be first sampled from actual distribution, and then the learner may decide whether to label it. Each unlabeled in-

stance is drawn one at a time from the data source, and the learner has to decide whether to label or discard it. So this procedure is called stream-based active learning.
In our survey of previous related works, we can find several ways to determine whether or not to label an instance. One implementation by Dagan and Engelson is to evaluate samples by some query strategy such as query-by-committee (QBC) algorithm. Each committee member is allowed to vote for the unlabeled candidate query, and the instance which has the most vote entropy is most likely to get labeled [7]. The second approach by Cohn et al. is to compute the explicit region of uncertainty, within which only query instances fall. In the most recent work [27], a classifier-ensemble based active learning frame work is implemented. An active learner randomly label a very small number of examples, and then selects a few additional examples and learns from the request to select which to label next, in order to maximize the prediction accuracy. The stream-based active learning scans through the data sequentially and makes query decisions individually. However, the pool-based active learning can evaluate the entire collection before selecting the best query, which is not adaptable in some real cases.

2.3 Cost-Sensitive Framework of Active Learning

As the previous discussion indicates that the method of active learning is well motivated in both pool-based data (also known as static data) and data stream field. However almost all of existing related works [27] [2] [7]before 2012[22] focus only on improving the accuracy instead of cost which may not always in accordance as real world issues. As a result two short comings of these implements.
First of all, since the cost of labelling data is not fully taken into consideration, in previous work like [21] concentrating on static-data and [27] exploring data-stream always select a fixed number of data to be labelled. The fixed size is defined by the maximum data we can label under certain situations. However we can easily find out that this simple well of candidate data amount selecting is sometimes wasteful. For an instance if we can acquire a satisfying result by just labelling 3% of total data, why should we continuously label 5% of the data though this mission is possible with current resources?
Furthermore, in machine learning field we call the Bayes classifier which only takes minimum error-rate instead of other facts as na?ve Bayes classifier that is not acceptable under many real world circumstances. For the two kinds of misclassification lost in binary classifying problems are different as the example of credit card record discussed above, the amount of respective issues named false-positive (fp) and false negative (fn) as well as their quantities (N_{fp},N_{fn}) are ought to be considered. Thus, lost functions shown below will be created[22].

$$C_p = N_{fp} \times fp + N_{tp} \times tp \tag{2}$$

$$C_n = N_{fn} \times fn + N_{tn} \times tn \tag{3}$$

Where C_p, C_n donates the cost of positive and negative class lost. However, as we all know, the unbalance of cost will cause severely unbalanced classification result due to the issue that we can classify all sample into the class whose misclassification lost is relatively low. So a Laplace correction $\lambda = |N_{ft} - N_{fn}|$ and $k = N_{ft} + N_{fn}$ [4],[8]is introduced to avoid that kind of situation.And the final cost of each are given below:

$$C_p = \frac{C_N + \lambda}{C_N + C_P + k} \times FP \tag{4}$$

$$C_p = \frac{C_P + \lambda}{C_N + C_P + k} \times FN \tag{5}$$

Though there are some other kinds of lost like CPU cost and increasing of time complexity, we do not count these facts to simplify the problem just as Sheng, V.S.[22] does. And the combination of labelling cost and misclassification cost are defined as the actual cost [22].

2.4 New COWCE in stream data

According to the survey above, we can conclude that there are two main performance measures regarding to active learning, namely, prediction accuracy and total cost (labeling and misclassification cost). In Victor S. Shengs paper[22], he proposed a pool-based cost-sensitive active learning algorithm and compared it to previous active cost-sensitive learning methods. He not only consider minimizing misclassification costs but also utilized the cost-sensitive decision tree to choose the attribute with the maximum cost reduction. However, this algorithm is only utilized in static pool-based data, and cannot be applied to many real applications such as data streams with continuous volumes. In another paper (Active Learning From Stream Data Using Optima Weight classifier ensemble) written by ZHU et al[27], the author studied how to label important instances from data stream via active learning. They build a classifier ensemble and reduce its variance to improve the prediction accuracy. But they didn't consider cost which is a better performance measure in some special situations.
In this paper, we put forward a cost-oriented weight classifier ensemble (COWCE) learning algorithm which is applied to data stream. We utilize labeling and misclassification cost as performance measure to train classifier in one chunk. Then we ensemble classifier in recent chunks to consider collect learning experiences together. COWCE can easily adapted to real world data stream cases where labeling and misclassification cost is much more important to predicting accuracy, just like the credit card example mentioned above. And there is nobody who has already implemented this algorithm so far.

2.5 OC-Ensemble Learning

The algorithm of ensemble learning is established based on an assumption that the aggregation of a bunch of classifier will provide more diverse[15][14] prediction based on different information each one of them emphasizing on. However the combining of a set of weak classifiers does not necessarily make a stronger ever classifier. Thus we should choose the way of ensembling very carefully.
The common method of ensemble learning turns to minimum the variance after ensembling. A pool based classifier

Pseudo-code is shown below[14].

function $trainBayesianModelCombination(T)$
For each model, m, in the ensemble:
$weight[m] = 0$
$sumWeight = 0$
$z = -infinity$
Let n be some number of weightings to sample.
(100 might be a reasonable value. Smaller is faster.
Bigger leads to more precise results.)
for i from 0 to n - 1:
For each model, m, in the ensemble: // draw from a uniform Dirichlet distribution
$v[m] = -log(randomUniform(0, 1))$
Normalize v to sum to 1
Let x be the predictive accuracy (from 0 to 1) of the entire ensemble, weighted
according to v, for predicting the labels in T.
Use x to estimate $logLikelihood[i]$. Often, this is computed as
$logLikelihood[i] = |T| * (x * log(x) + (1 - x) * log(1 - x))$,
where |T| is the number of training patterns in T.
If $logLikelihood[i] > z$: // z is used to maintain numerical stability
For each model, m, in the ensemble:
$weight[m] = weight[m] * exp(z - logLikelihood[i])$
$z = logLikelihood[i]$
$w = exp(logLikelihood[i] - z)$
For each model, m, in the ensemble:
$weight[m] = weight[m] * sumWeight/(sumWeight + w) + w * v[m]$
$sumWeight = sumWeight + w$
Normalize the model weights to sum to 1.

However, this kind of ensembling based on the hypothesis that both misclassification error cost are the same. To fulfill the cost-oriented class task we should take cost into regard during the process that firstly introduced by [6]
Also we must implement our method to data stream make the classifier extracted from each data trunk work together.

3. STATES REPORT

3.1 Machine Learning And SVM Introduction

In machine learning, we need to train the classifier to divide the given data points into two classes according to their attributes, and the final goal is to decide which class a new data will be distributed into. Support vector machine (SVM) is a supervised learning method wildly applied in classification and regression analysis. The data point in SVM is viewed as a p-dimensional vector, and a (p-1)-dimensional hyperplane will be considered to separate the points into two classes. We can decide the best hyperplane by the largest margin between the two classes, and the distance from the hyperplane to the nearest data point on each side is maximized. The above SVM is a linear classifier, as shown in figure 1.
For nonlinear situations, we can apply the kernel function to maximum-margin hyperplanes [1]. We choose a kernel function to map the input points to a high-dimensional feature space, in which we build the optimal classifier hyperplane

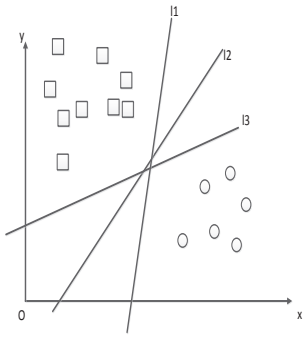

Figure 1: The SVM Classification Illustration

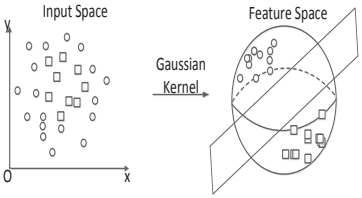

Figure 2: The SVM Classification Illustration

and this hyperplane may be nonlinear in the original input space.
In this paper, we adopt Gaussian radial basis function , as shown in figure.2.

3.2 Basic Introduction of Crossvalindation

Before we implement labeling one chunk data with active learning method, we apply cross-validation to this static data set in order to assess the stability of data model and estimate the accuracy of this predictive model will perform. Basic theory of cross-validation is that we can divide data set into a few parts. We specify most of parts as training sets which are selected for classifier training and one of them as validation set to test the data model so that we can evaluate the performance of the classifier.
There are some types of cross-validation. One of most widely-used is called K-fold Cross Validation. In K-fold cross-validation, the original data set is randomly divided into K equal size subsets. Of the K subsets, a single subset is used as the validation data for testing the model, and the remaining K-1 subsets are used as training data. The cross-validation process is then repeated K times(the folds), with each of the K subsets used exactly once as the validation data, the K results from the folds then can be averaged to produce a single estimation [13]. In our experiment, we choose 5-fold cross-validation as figure.3 shows.

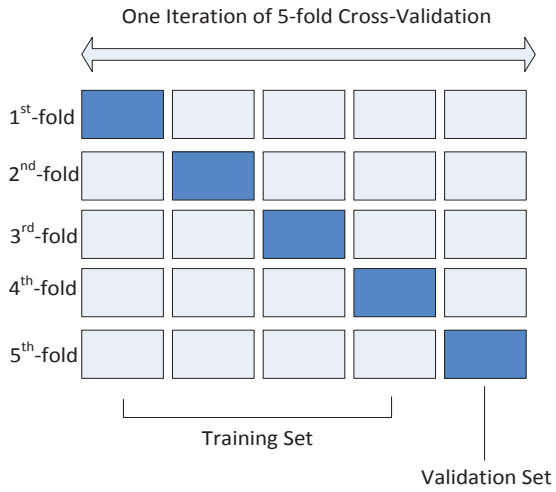

Figure 3: 5-Fold-Crossvalindation Illustration

3.3 Testing Data And Analysis Platform

We used a open source framework called MOA(Massive Online Analysis) based on JAVA platform [3], which is a very popular data-mining tool, to generate the data stream. At this case, we choose to use a RBFGenerator-API to generator 990000 data with 10 attributes and 2 classes. Later the 99000 data will be divided into 99 trunks each of which contains 10000 samples. Within the 10000 samples, there are 5000 positive instance as well as 5000 negative instance. We will take all of this 10000 samples into calculation in section 3.4 and the first experiment of 3.5.

Under the hypothesis that the samples with class which causes massive lost will not appear as frequently as the other class′s instances. So we make the data stream ratio to be unbalanced about 1:5 in quantity.

To analysis the data we used the libsvm platform [5] as well as weka data-mining tool[9] platform created by the same author of MOA with JAVA.

3.4 The Advantage of Active Learning

As above, even though K-fold cross-validation is a common method of accuracy measuring, we need to know that there is a significant disadvantage that is high computational cost. The reason is that each time when we need to model, the number of training data is nearly all of the original data set which is prohibitively expensive and undesirable (we will discuss it later). What if we find a method to solve this problem? The answer is active learning.

Initially, we randomly choose 1% data in sample data chunk as training set and data modeling and use other 99% data set as validation set to test the model. According to the

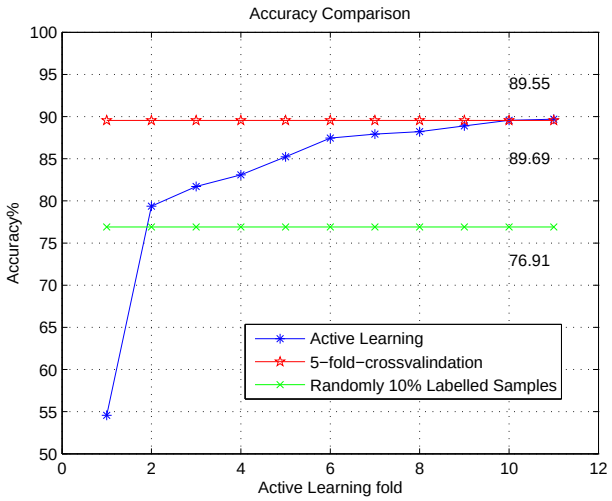

Figure 4: Accuracy Comparison Between 5-fold-crossvalindation,active-learning and 10% randomly selected sample

figure 4, we see that accuracy is nearly 55%. After that, we need to use active learning method based on SVM to select another 1% data which is the most difficult part to classify. It is reasonable for us to think that those points which are closest to classifier are the most difficult part. In other words, the largest amount of information are represented as those points which are closest to hyper plane(around the green bond of figure.5). We dont need to consider those points far away from the hyper plane such as the point (circled by blue and black mark on figure.5) in figure 5 since they are very easy to be classified. At this time, we have labeled 2% sample data and the accuracy is nearly 80% as figure 5 shows. After we labeled 10% sample data using active learning method, we can figure out that the accuracy is 89.69% which is a little better than 5-fold cross-validation method.And we can find out that the FN as well as FP also declined with the processing of active learning as figure.6 shows

In figure 4, we can also find that if we randomly choose 10% data as labelled samples, the accuracy is only 76.91% which is lower than active learning method.

3.5 Introduction Of Cost Frame

Previously we have discussed about the necessity of introducing cost as a performance measure of active learning. Misclassification and labeling are two main sources of cost, and false positive and false negative in binary classification may as well be quite different in cost regarding to some real world cases. However, if we consider the cost as the main performance measure of active learning, we must make sure that the accuracy lost is not beyond our limit. So in order to verify that after introducing cost we can hold the relatively high accuracy, we firstly apply it into 5-fold cross-validation. The experiment result is shown in figure 7.

As is shown in figure 7, the blue bars stand for the original 5-fold cross-validation whose performance measure is accuracy. The brown bars stand for the new method into which cost is applied. As we can see, total cost in new method is

Table 2: Active Learning Performance - naive

Fold	0(random)	1	2	3	4	5	6	7	8	9	10
Accuracy	54.56	79.37	81.707	83.09	85.22	87.46	87.92	88.22	88.9	89.57	89.69
TP	2823	4078	4189	4244	4387	4315	4201	4186	4141	4098	4077
TN	2633	3700	3736	3733	3709	3906	3976	3930	3949	3963	3906
FP	2427	1116	1027	989	968	720	599	596	529	470	486
FN	2017	906	748	634	436	459	524	488	481	469	431
SUM	9900	9800	9700	9600	9500	9400	9300	9200	9100	9000	8900

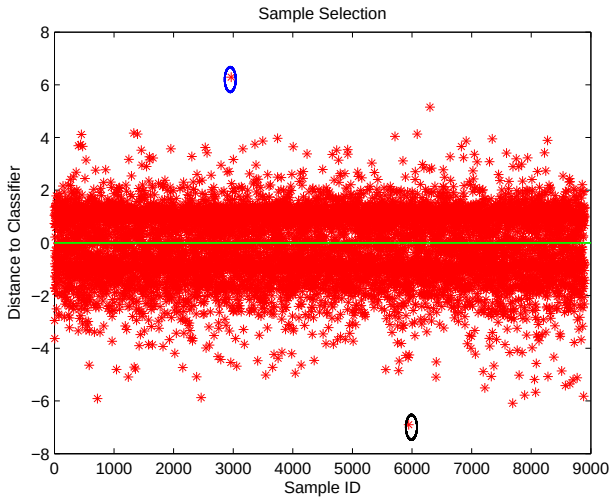

Figure 5: Illustration of Data Selection

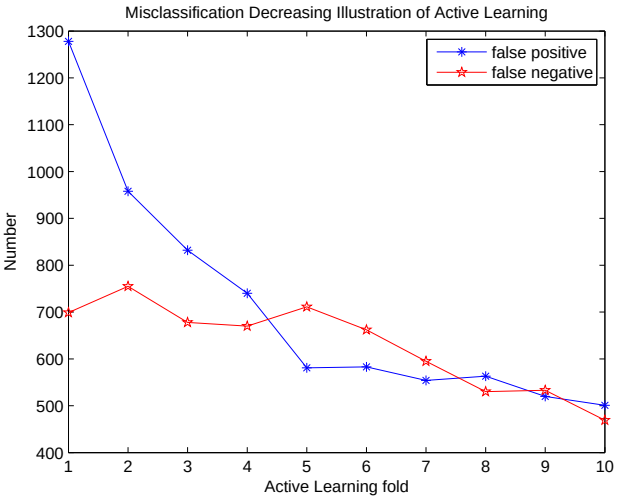

Figure 6: The Decline of Misclassification

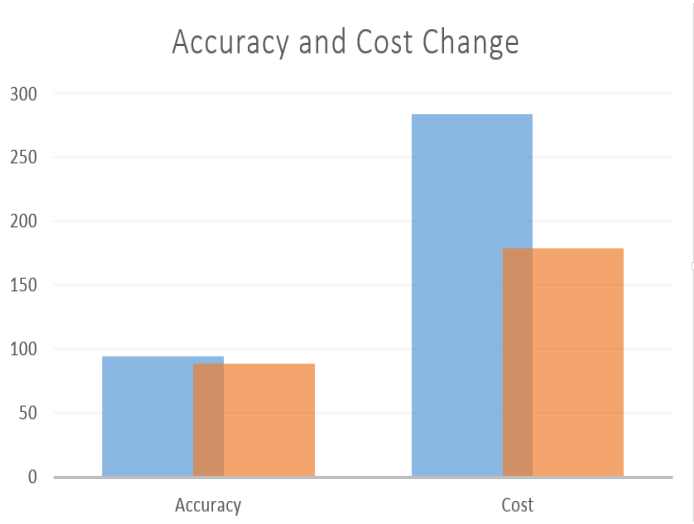

Figure 7: Illustration of Data Selection

considerably decreased, while the accuracy almost remains the same. This experiment result shows that it is a feasible way to apply cost into SVM, and we can remain a relatively high accuracy when we minimize the total cost.

Another important factor that humbles the accuracy criterion is the unbalanced dataset we choose. As we mentioned it previously in section 3.3 that the appearance rate of abnormal samples are relatively low(5:1) in our experiment. This will cause a problem of deceptive prosperous result. Namely, we class all samples to the normal,which means not a fraud transaction in credit card issue, will lead us to a accuracy of 5/6 that is 83.33%. It is a pretty high performance if we only focus on the accuracy. However, as we all know, this result is meaningless. As we can see from table 2 and table 3 the randomly picked 1% samples′ accuracy raises dramatically due to the change of dataset. To avoid this kind of judgemental mistake we must introduce cost into our algorithm. Since the misclassification cost of abnormal events will result in a huge lost, classify every instance as normal will be unacceptable due to its magnificent lost, thus eliminating the deceptive prosperous result.

When we talk about active learning in the Cost-Sensitive framework, we should know which kinds of cost we need to consider as the evaluation criterion of the active learning. In our experiment, we assume that the total cost consists of labeling and misclassification cost. Furthermore, misclassification cost can be divided into two parts. One is false positive cost, the other is false negative cost. Consider our credit card transaction case, false positive cost means the cost that we classify credit card fraud as a normal transaction, which

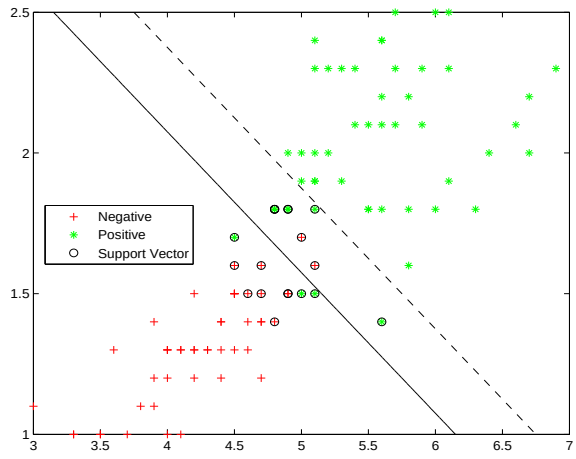

Figure 8: Illustration of Classifier Shift

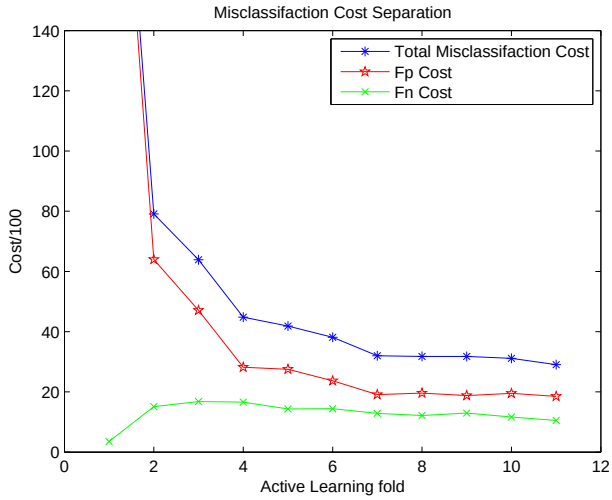

Figure 9: The Total Lost, Fn Lost And Fp Lost

is a great danger and may become a big loss to the credit user. So the cost of false positive can be considered high. False negative cost means that we classify a normal transaction as credit card fraud, so it is relatively low. Meanwhile, no other costs (such as computational resources and memory cost) are considered in this paper. Therefore, these two kinds of cost are on the same scale (such as dollars) and active learning becomes the minimization of the total cost. For simplicity, we use constant value to represent these two kinds of cost. For example, we define the cost of labeling an object is 1.5, false positive cost is 10 and false negative cost is 1. In a practical way, if we use 5-fold cross-validation method, the more numbers of sample data means the more cost so that we need active learning method instead to solve this problem.

Since the data point is ten-dimensional, we cannot draw a hyper plane of nine-dimension. For simplicity, we draw a

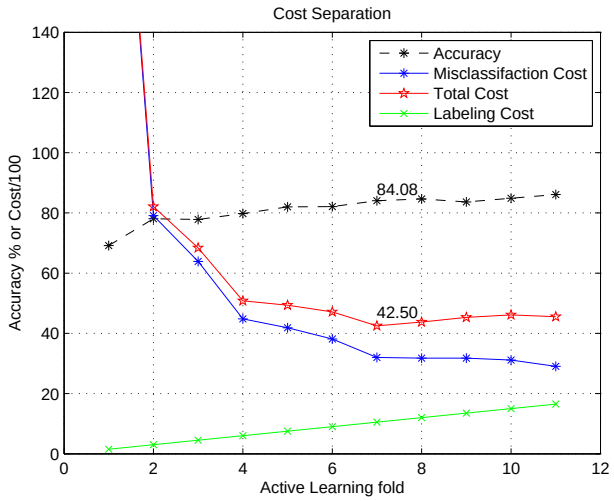

Figure 10: The Cost And Accuracy Together Using PoolCA In one Trunk

one-dimensional hyper plane to show the classification of data points. Before we apply cost in active learning, we can get a classifier showed in bold line in figure 8. The red points stand for negative class and the green points stand for positive class. However, we can figure out that some of red points are classified into positive class, which are called false positive. Moreover, some green points are classified into negative class which are called false negative. Although the classifier can divide the points based on the best accuracy, those false positive points can be fatal because of their high cost. So in order to minimize the total cost, decreasing the false positive points is an effective way. After we apply cost in active learning, the newly trained classifier shifts to right, as is showed in dotted line in figure 8. We can see that even though the number of false negative points is increasing, the number of false positive is decreasing just as what we expect. This is the way how we handle the total cost.

Figure 9 shows the decreasing of misclassification cost when active learning goes on. As we can see, false positive cost decreases dramatically, while false negative increases at the beginning and remains almost the same afterwards. Since we define that the cost of false positive is much higher than that of false negative, the total cost will decrease in general. More ideally, both of them will decrease in general if the model is further optimized. Figure 10 shows the change of both the cost and accuracy. We also take labeling cost, showed as green line, into consideration, which is a linear increase because it is determined only by how many examples we will label to train the classifier. The blue line is the change of misclassification cost, and the total cost is showed as the red line, which has a minimal value of 42.5. The black line shows the accuracy changes, which is gradually increased and is around 80 percent. Corresponding to the minimal value of total cost of 42.5, the accuracy of active learning is 84.08, which is a little less than 89.69 discussed in figure 4 and is acceptable. Until now we can conclude we have successfully applied cost in active learning in one trunk of data. The following work should be how we can ensemble the classifiers trained in each trunk to build the final classifier of the data stream.

Table 3: Active Learning Performance - cost

Fold	0(random)	1	2	3	4	5	6	7	8	9	10
Accuracy	69.17	78.04	77.83	79.77	82	82.11	84.08	84.64	83.66	84.89	86.09
TP	4681	3467	3251	3216	3393	3333	3429	3446	3315	3394	3455
TN	2168	4182	4299	4443	4398	4387	4391	4342	4299	4247	4208
FP	2702	640	471	282	275	237	191	196	188	195	185
FN	350	1512	1680	1660	1435	1444	1290	1217	1299	1165	1053
COST	27520	8212	6840	5080	4935	4714	4250	4377	4529	4615	4553

4. FINAL REPORT

4.1 Ensemble learning Introduction

Ensemble learning is a machine learning paradigm where multiple learners are trained to solve the same problem [16]. Actually, the reason why multiple classifiers ensemble became a hot topic in recent years are mainly two reasons. On one hand, the generalization of ability of an ensemble is usually much stronger than that of base learners (In our project, base learners, also referred as weak learners, are generated from training set by a cost-oriented active learning algorithm). This is according to the facts that at the end of 1980s an applied research conducted by Hansen and Salamon [10]showed that the combination of a set of classifiers are often more accurate than predictions made by the best single classifier. Also, a theoretical research conducted in 1990, where Schapire [20] proved that weak learners can be boosted to strong learners. On the other hand, stream data provides solid foundation for constructing ensemble successfully since a number of base learners are produced, which can be generated in a sequential style where the generation of a base learner has influence on the generation of subsequent learners. We make fully use of stream data and the storage capacity is very small.

4.2 The LSE-based weighting

Based on the analysis above, we firstly used weighted average and ExpectationŰmaximization (EM) algorithms to ensemble classifiers. The proposed framework consists of 3 major steps: 1)initialization; 2)labeling new instances; 3) calculating optimal weight values for ensemble learning. In step 1) we firstly initialize some specific weight values for those classifiers. Then the loop between 2) and 3) repeats with one another based on the results of each other. The expected goal is that the new labeling and the ensemble weight updating are beneficial to each other and the EM logics are as follows:

1) E-step: Use a set of weighted values for the classifier ensemble to label the instances with minimized misclassification and labeling cost.

2) M-step: Use the newly labeled instances to calculate the optimal weight values for the classifier ensemble, so that the overall prediction cost will be minimized. Figure 11 shows the outcome of the ensemble learning by weight average. We generate 99 chunks of data, with around 6000 data in each chunk. We use 10 classifiers in each of the 10 latest chunks to ensemble the classifier in the current chunk. The red line shows the total cost determined by the classifier generated by Pool-based Cost-Sensitive Active Learning (PoolCA) in one chunk. The blue line shows the total cost determined by the classifier ensemble. As we can see, the outcome isnt stable and not satisfying. The cost by ensemble learning is nearly the same with that by PoolCA.

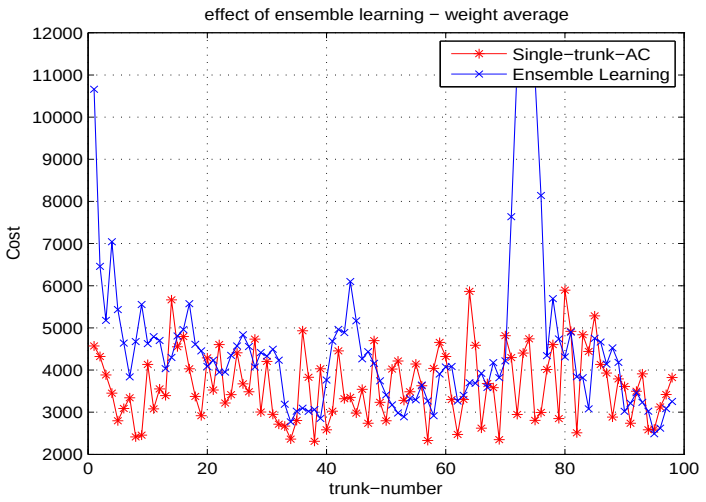

Figure 11: Cost Of Sub-Classifier and Cost of Ensemble Classifier Using The LSE-based weighting

4.3 The double-layer hierarchical combining

The general idea is to use another SVM to aggregate the outputs of several SVMs in the SVM ensemble. First we put a test data set into the 10 SVMs from the latest 10 chunks and generates the output matrix. Then we use this output matrix to train the new classifier in the current chunk. So the combination consists of a double layer of SVMs hierarchically where the outputs of several SVMs in the lower layer feed into a super SVM in the upper layer.

Let f_k (k = 1,2,Ě,K) be a decision function of the kth SVM in the SVM ensemble and F be a decision function of the super SVM in the top level. The decision of the SVM ensemble f_{SVM}(x) for a given test vector x is [12]:

Figure 12 shows the outcome of the ensemble learning by Top-level com4bining algorithm. We generate 99 chunks of data, with around 6000 data in each chunk. We use 10 classifiers in each of the 10 latest chunks to ensemble the classifier in the current chunk. We find that the cost generated by single chunk PoolCA and ensemble learning is almost the same. So, due to the little improvement, we need to find a better way of ensembling the classifier.

4.4 Majority Voting

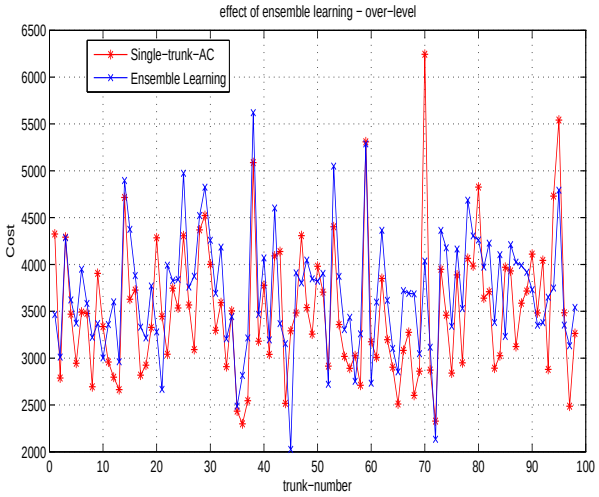

Figure 12: Cost Of Sub-Classifier and Cost of Ensemble Classifier Using double-layer hierarchical combining

When it comes to ensemble learning it means one kind of ensem4ble methods which constructs a set of classifiers and then classify new data by taking a vote of their predictions. The method of ensemble learning we use is majority voting. This is a very effective and common approach for combining several SVMs which are used to classify data for labeling [19]. The result of majority voting will show why this method of ensemble can perform better than single classifier and other methods.

In our project, we have most recently ten individual classifiers as committee members. We also have a huge number of data which is divided into 1,000 data chunks. Each chunk has nearly 6,000 data records. When we classify data records, we need to take other committee membersŠ information into consideration to predict and label. In other words, other classifiers votes are very important to determine the prediction of label. In our situation, we classify fraud transaction into positive and normal transaction into negative. We also set different threshold values which determine that normal transaction will be classified into negative only if it amounts to specific percentage of committee membersŠ votes. The effect of majority voting as figure 13 and figure 14 shows.

Initially, from figure 13 and figure 14, we can conclude that SVM ensemble can improve the classification performance greatly than using a single SVM since the combination of several SVMs will expand the correctly classified area incrementally. Furthermore, we can figure out that different threshold values result in different cost. It is obvious if we compare these two figures in detail. For figure 13, if the number of votes for fraudulent transaction are equal or greater than 30% of the total number, the data record will be classified into positive (fraudulent transaction). However, in figure 14, if the number of votes for fraudulent transaction are equal or greater than 40% of the total number, the data record will be classified into positive (fraudulent transaction). That is, the data record is easier to be classified into fraudulent transaction in scenario 1 than scenario 2 which means more probably the true fraudulent transaction will be classified into false normal transaction so that the

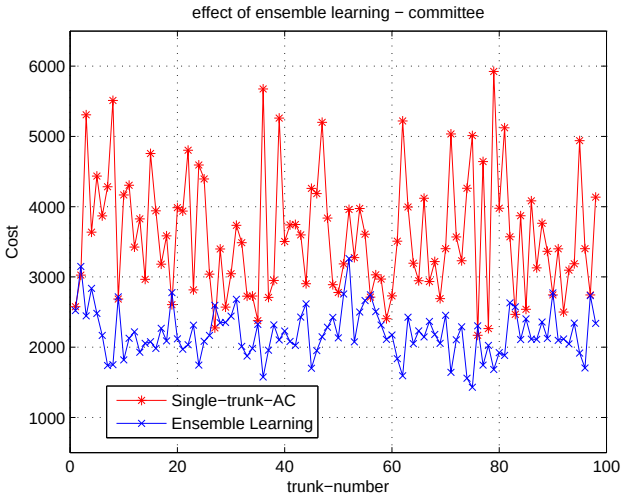

Figure 13: Cost Of Sub-Classifier and Cost of Ensemble Classifier Using Majority Voting with 70% Agree

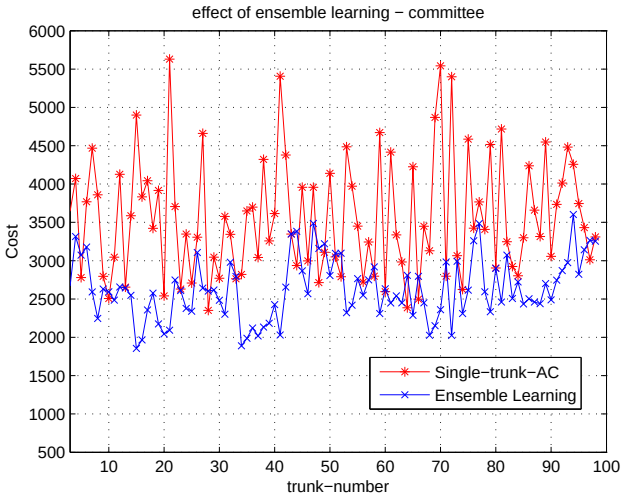

Figure 14: Cost Of Sub-Classifier and Cost of Ensemble Classifier Using Majority Voting with 60% Agree

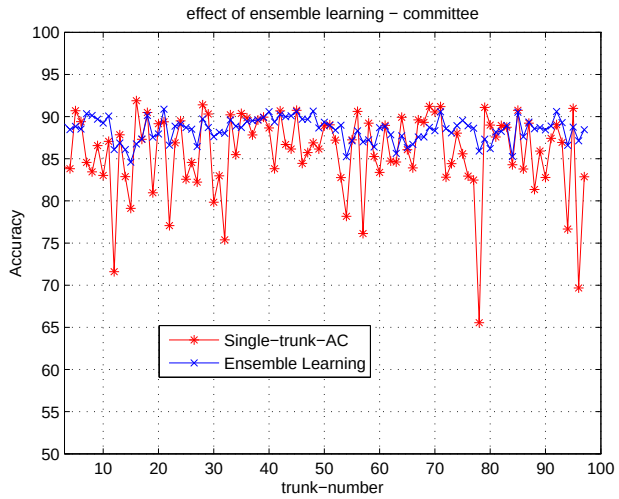

Figure 15: Cost Of Sub-Classifier and Cost of Ensemble Classifier Using Majority Voting with 70% Agree

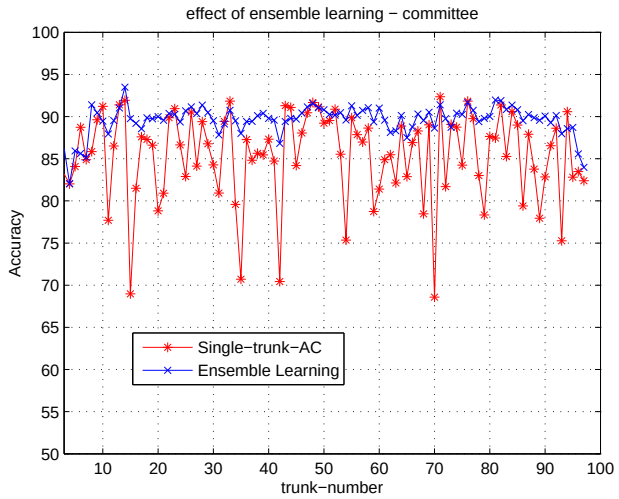

Figure 16: Cost Of Sub-Classifier and Cost of Ensemble Classifier Using Majority Voting with 60% Agree

cost is higher in scenario 2.

Figure 15 and figure 16 shows the accuracy in a single SVM and the combination of several SVMs. We can find that the accuracy is also better if we implement ensemble learning.

4.5 Summary

Generally speaking, the combination of a set of weak classifiers does not necessarily make a stronger ever classifier and guarantee a better performance unless we choose the way of ensembling very carefully. In our project, we tried three different kinds of ensemble methods and figured out that only the majority voting ensemble method is a good choice for our problem. Also, this paper fill the gap in the field of machine learning that is cost-oriented active learning on stream data. Due to the limited resource that we can access, the lacking of test from huge realistic big data may undermine the reliability and make the algorithm vulnerable. Moreover, since the capability of our personal computer is limited, we are not able to try deep learning and some other modern models. Therefore, we hope we can implement those new algorithms in the future.

5. REFERENCES

[1] A. Aizerman, E. M. Braverman, and L. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.

[2] L. Atlas, D. Cohn, R. Ladner, M. A. El-Sharkawi, and R. Marks II. *Training connectionist networks with queries and selective sampling*. Morgan Kaufmann Publishers Inc., 1990.

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

[4] C. Blake and C. J. Merz. {UCI} repository of machine learning databases. 1998.

[5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[7] I. Dagan and S. P. Engelson. Committee-based sampling for training probabilistic classifiers. In *ICML*, volume 95, pages 150–157, 1995.

[8] I. J. Good. *The estimation of probabilities: An essay on modern Bayesian methods*, volume 30. MIT press Cambridge, MA, 1965.

[9] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[10] L. K. Hansen and P. Salamon. Neural network ensembles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(10):993–1001, 1990.

[11] S. Huang. An active learning method for mining time-changing data streams. In *Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on*, volume 2, pages 548–552, 2008.

[12] H.-C. Kim, S. Pang, H.-M. Je, D. Kim, and S. Yang Bang. Constructing support vector machine ensemble. *Pattern recognition*, 36(12):2757–2767, 2003.

[13] R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, volume 14, pages 1137–1145, 1995.

[14] P. S. A. Krogh. Learning with ensembles: How over-fitting can be useful. In *Proceedings of the 1995 Conference*, volume 8, page 190. The MIT Press, 1996.

[15] L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003.

[16] S. Z. Li and A. K. Jain. *Encyclopedia of Biometrics:*

I-Z., volume 2. Springer, 2009.

[17] X. Li and X. Xiao. Study on the combination weighting method of hybrid multiple attribute decision-making. In *Grey Systems and Intelligent Services (GSIS), 2011 IEEE International Conference on*, pages 561–565, 2011.

[18] P. Liu, Y. Wang, L. Cai, and L. Zhang. Classifying skewed data streams based on reusing data. In *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, volume 4, pages V4–90–V4–93, 2010.

[19] P. Melville and R. J. Mooney. Diverse ensembles for active learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 74. ACM, 2004.

[20] R. E. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[21] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 2010.

[22] V. Sheng. Studying active learning in the cost-sensitive framework. In *System Science (HICSS), 2012 45th Hawaii International Conference on*, pages 1097–1106, 2012.

[23] B. Sun, W. Ng, D. Yeung, and J. Wang. Localized generalization error based active learning for image annotation. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 60–65, 2008.

[24] T. Tsutaoka and K. Shinoda. Acoustic model training using committee-based active and semi-supervised learning for speech recognition. In *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1–4, 2012.

[25] R. Vallim, A. C. P. L. F. De Carvalho, and J. Gama. Data stream mining algorithms for building decision models in a computer role-playing game simulation. In *Games and Digital Entertainment (SBGAMES), 2010 Brazilian Symposium on*, pages 108–116, 2010.

[26] B. Wiese and C. Omlin. *Credit card transactions, fraud detection, and machine learning: Modelling time with LSTM recurrent neural networks*. Springer, 2009.

[27] X. Zhu, P. Zhang, X. Lin, and Y. Shi. Active learning from stream data using optimal weight classifier ensemble. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(6):1607–1621, 2010.