Naïve Bayes Classification Ensembles to Support Modeling Decisions in Data Stream Mining

Patricia E.N. Lutu
Department of Computer Science
University of Pretoria
Pretoria, South Africa
Patricia.Lutu@up.ac.za

***Abstract*—Data stream mining is the process of applying data mining methods to a data stream in real-time in order to create descriptive or predictive models. Due to the dynamic nature of data streams, new classes may emerge as a data stream evolves, and the concept being modeled may change with time. This gives rise to the need to continuously make revisions to the predictive model. Revising the predictive model requires that labeled training data should be available. Manual labeling of training data may not be able to cope with the speed at which data needs to be labeled. This paper proposes a predictive modeling framework which supports two of the common decisions that need to be made in stream mining. These decisions are: (1) determining when model revision should be performed and (2) deciding which newly arrived instances should be used as training data. The framework consists of an online component and an offline component. The online component uses Naïve Bayes ensemble base models to make predictions for newly arrived data stream instances. The offline component consists of algorithms to combine base model predictions, determine the reliability of the ensemble predictions, select training data for new base models, create new base models, and determine whether the current online base models need to be replaced.**

I. Introduction

Data stream mining is defined as the process of applying data mining methods to a data stream in real-time in order to create descriptive or predictive models for the process that generates the data stream [1], [2], [3]. Due to the dynamic nature of data streams, the concept being modeled may change abruptly or gradually with time [4], [5], [6]. The changes in the concept being modeled give rise to the need to continuously make revisions to, or completely rebuild the predictive model when these changes are detected. Rebuilding the predictive model requires that labeled training data should be available. Data labeling for stream mining needs to be a fast process since stream data may arrive at a very high speed. Traditional methods of labeling training data may not be able to cope with the speed at which data needs to be labeled.

Given the foregoing discussion, various decisions need to be made for the predictive modeling process in stream mining [7]. Two of these decisions are: (1) determining when model revision should be performed and (2) deciding which newly arrived instances should be used as training data. This paper proposes a predictive modeling framework for stream mining based on Naïve Bayes ensemble classification [8]. The framework consists of an online component and an offline component. The online component uses Naïve Bayes ensemble base models to make predictions for newly arrived data stream instances. The offline component consists of algorithms to combine base model predictions, determine the reliability of the ensemble predictions, select training data for new base models, create new base models, and determine whether the current online base models need to be replaced. The main objective of this framework is to remove the need for manual labelling of training instances. A secondary objective is to detect when model revision should be performed.

Three measures for assessing the performance of the ensemble base models are proposed in this paper. These measures are: *Certainty*, *Reliability*, and *Incoherence*. C*ertainty* measures the frequency that all base models in the ensemble have predicted the same class. *Incoherence* measures the frequency that each base model in the ensemble has predicted a different class from the other base models. *Reliability* measures the frequency that one class is predicted by the majority of base models in the ensemble. Experiments were conducted to assess the usefulness of these measures in supporting decisions for the automated selection of new training data. The experimental results reported in this paper demonstrate that the proposed measures provide useful information for decision making and that the proposed framework has a high potential to produce predictive models with practical value. The rest of this paper is organised as follows: Section II provides the background to the reported research. Section III presents the proposed stream mining framework. The experimental results are presented in Section IV. Section V concludes the paper.

II. Background

A. Challenges in Stream Mining

One major challenge for mining data streams is due to the fact that it is infeasible to store the data stream in its entirety. This problem makes it necessary to select and use training data that is not outdated for the mining task. The second challenge for stream mining is due to the phenomenon of concept drift, which is defined as the gradual or rapid changes in the concept that a mining algorithm attempts to model [1], [2], [3]. Given these challenges, there is a need to continuously monitor model performance and revise the model when the performance degrades. The third challenge is due to the fact that for predictive classification modelling there is a need to rapidly and continuously provide training

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.57

data which consists of instances that are labelled with the classes.

One approach to selecting training data for mining data streams is called the sliding window approach. A sliding window, which may be of fixed or variable width, provides a mechanism for limiting the data used for modeling to the most recent instances. The main advantage of this technique is to prevent stale data from influencing the models obtained in the mining process [5], [6]. Two main problems with this approach are that firstly, for predictive modeling, there is an in-built assumption that labelled training data is rapidly available. Secondly, the predictive model needs to be continuously recreated as the window slides. Data stream instances do not typically arrive in an independent and identically distributed (iid) fashion. It is possible for instances of one class to arrive over a prolonged period of time. When this is the case, it may become infeasible to employ the sliding window approach, as the model could end up being trained on instances of one class only! A second approach to stream mining is to employ ensemble classification. Ensemble models for stream mining resolve the problems created by the sliding window approach by creating a new base model only when a new batch of labelled instances arrives and keeping base models that are trained on both old and new instances.

Masud et. al [9], Zhu et. al [10] and Zhang et. al [11] have all observed that, for stream mining, manual labelling of data is a costly and time consuming exercise. In practice it is not possible to label all stream data, especially when the data arrives at a high speed. It is therefore common practice to label only a small fraction of the data for training and testing purposes. Masud et. al [9] have proposed the use of ensemble classification models based on semi-supervised clustering, so that only a small fraction (5%) of the data needs to be labelled for the clustering algorithm. Zhu et. al [10] have proposed an active learning framework for solving the instance labelling problem. The main challenge of active learning is to identify the most informative instances that should be labelled in order to achieve the highest accuracy.

Active learning is a branch of machine learning concerned with the automated selection of the most useful instances that should be manually labelled by a human expert [12]. Typically, these are instances that are located in the decision boundaries of the instance space. Settles [12] has observed that even though active learning provides a practical solution to the cost reduction for instance labelling, it suffers from two major weaknesses. The first major weakness is due to the fact that the training instances are a biased distribution and not an iid sample which represents the underlying natural density of the available data. The second major weakness is due to the computationally intensive algorithms for active learning. For each instance that is processed by the algorithm, a value must be computed for the measure of informativeness, measure of disagreement, or some other measure.

B. Ensemble Classification for Stream Mining

Several ensemble classification methods for stream mining have been reported in the literature. These methods include the use of All-Classes-At-once (ACA) base models (e.g. [13], [14]), the use of All-Versus-All (AVA) base models (e.g. [15]) or the use of OVA base models (e.g. [16]). The main objective of ACA ensembles for stream mining is to (1) avoid overfitting, (2) avoid the use of a very limited amount of training data as is the case for the sliding window model, and (3) reduce the computational effort of revising the whole model when concept change/drift is detected [13], [14]. Examples of ACA ensemble frameworks that have been reported in the literature are the streaming ensemble algorithm (SEA) [17], the accuracy-weighted ensemble (AWE) [13], and the dynamically weighted majority (DWM) ensemble [14]. In the context of active learning, ensemble models have also been used in the Query-by-Committee (QBE) scheme to determine those instances that require manual labeling [12].

III. PROPOSED STREAM MINING FRAMEWORK

As stated above, the main objective for the research reported in this paper was to study methods for eliminating the need for manual labelling of training instances. A secondary objective was to study methods for determining when model revision should be performed. The approach that was adopted was to create a stream mining model consisting of two components: an online component and an offline component. The online component uses Naïve Bayes ensemble base models to make predictions for newly arrived data stream instances. The offline component consists of algorithms to combine base model predictions, determine the reliability of the ensemble predictions, select training data for new base models, create new base models, and determine whether the current online base models need to be replaced. This section provides a description of the online and offline components of the proposed framework.

A. Naïve Bayes Classification

Naïve Bayes classification has been reported in the literature as one of the 'ideal' algorithms for stream mining, due to its incremental nature [18]. The Naïve Bayes classifier assigns posterior class probabilities for the query instance $\boldsymbol{x}$ based on Bayes theorem. The training dataset for a classifier is characterised by d predictor variables $X_1,..,X_d$ and a class variable C. The set of n training instances is denoted as $\{(\mathbf{x},c_i)\}$ where $\boldsymbol{x} = (x_1,...,x_d)$ are the values of a training instance and $c_i \in \{c_1,...,c_K\}$ are the class labels. Given a new query instance $\boldsymbol{x} = (x_1,...,x_d)$ Naïve Bayes classification involves the computation of the score for each class defined as

$$Pr(C = c_j \mid \boldsymbol{X} = \boldsymbol{x}) \propto$$

$$Pr(C = c_j) \prod Pr(X_i = x_i \mid C = c_j) \qquad (1)$$

For zero-one loss classification, the class c_j with the highest score is selected as the predicted class. For categorical features, the quantities $Pr(C = c_j)$ and $Pr(X_i = x_i \mid C = c_j)$ are estimated from the training data. Naïve Bayes (NB) classification was selected for the proposed framework for two reasons. Firstly, NB classification is a probabilistic classification framework

based on the estimation of probability values from the data. This makes it difficult to justify the use of active learning as a suitable training instance selection method for NB classification. Secondly, model creation is a simple and fast activity. A single contingency table of counts for the feature values and class labels provides all the data needed for the computation of the probabilities in equation (1) as well as the selection of the relevant features for prediction [19].

B. Components of the Framework

The online component uses Naïve Bayes ensemble base models to make predictions for newly arrived data stream instances. Fig. 1 depicts the online component of the framework. The sub-components M1, M2 and M3 represent the Naïve Bayes base models. When a new data stream instance $\boldsymbol{x}_q$ arrives, each of the base models provides a prediction for the instance. The instance as well as the predictions are written to a disk file for further processing by the offline component. This ensures that the online component can operate at a very high speed.

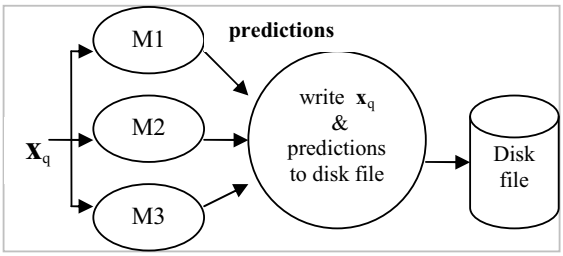

Fig. 1. The online components

The offline component consists of five algorithms. Fig. 2 depicts this component of the framework. The first algorithm is the combination algorithm which is used to determine the class predicted by the ensemble. The majority rule is used for the selection. The second algorithm is used to assign each prediction to one of three categories as specified in Table I. The third algorithm is used to determine whether an instance should be selected for the training data of a new base model, based on the prediction category for the instance. If the prediction category is *certain* or *reliable*, the instance is selected. The fourth algorithm is used to create new base models in the form of contingency tables for Naïve Bayes classification. The fifth algorithm is used to determine whether there is a need to revise the online ensemble by replacing one or more of the base models. The decision is based on the proportions of the prediction categories at the end of a time period. Three measures are used to assess the prediction performance of the ensemble. The measures are *Certainty*, *Reliability*, *Incoherence*, and are defined as follows:

$$Certainty = count(certain) \,/\, count(all\ categories) \quad (2)$$

$$Reliability = (count(certain) + count(reliable)) \,/\, count(all\ categories \quad (3)$$

$$Incoherence = count(incoherent) \,/\, count(all\ categories) \quad (4)$$

C*ertainty* measures the frequency that all base models in the ensemble have predicted the same class. *Incoherence* measures the frequency that each base model in the ensemble has predicted a different class from the other base models. *Reliability* measures the frequency that one class is predicted by the majority of base models in the ensemble. This is used as a surrogate measure of accuracy. High levels of *Certainty* and *Reliability* indicate that the current online model is providing high predictive performance. A low level of *Certainty* and high level of *Incoherence* indicate that the current online model is providing poor predictive performance, and needs to be changed.

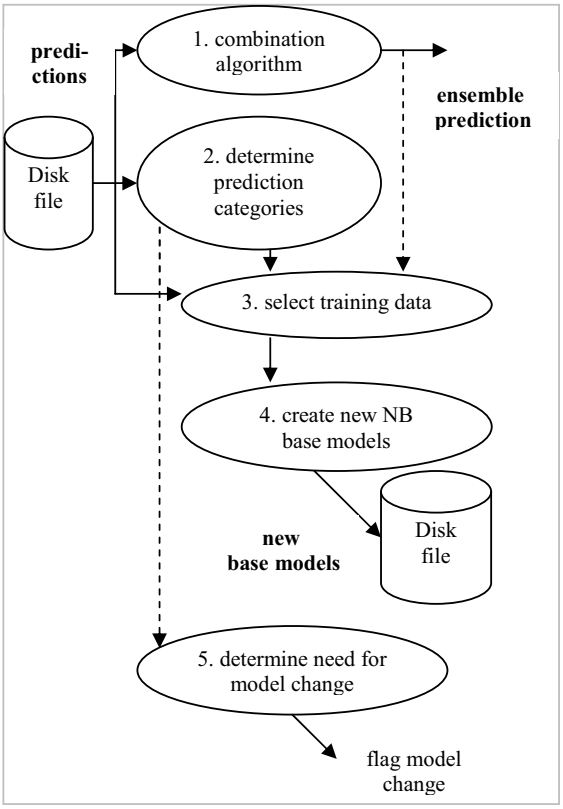

Fig. 2. The offline component

TABLE I. PREDICTION CATEGORIES

Category	*Description*	*meaning*
certain	all models agree	pred(M1) = pred(M2) = pred(M3)
reliable	most recent 2 models agree	pred(M2) = pred(M3) ≠ pred(M1)
	oldest model agrees with most recent model	pred(M1) = pred(M3) ≠ pred(M2)
	oldest 2 models agree	pred(M1) = pred(M2) ≠ pred(M3)
incoherent	no agreement between models	pred(M1) ≠ pred(M2) ≠ pred(M3)

C. Problems that are Solved by the Proposed Framework

It was stated in Section II-A that instance labelling by human experts is a slow and expensive activity. Even though active learning reduces the number of instances that require labelling by human experts, it results in the usage of training samples which have a biased probability distribution. This is problematic for probabilistic classifiers such as Naïve Bayes. The proposed framework makes it possible for iid training data to be obtained at low cost and very high speed without the need for labelling by human experts.

A second problem that is solved by the framework is the assessment of model performance. In the literature on predictive modeling for data stream mining (e.g. [7]) it is often stated that the accuracy of the predictive model is periodically tested to determine whether there is a need to revise the model. Measuring the accuracy of a model requires test data with class labels. Since the class labels for training and test data are determined by human experts, there is a high level of latency between the time when this data appears in the data stream and when it is used for testing. The proposed framework solves this problem by using a meaningful measure of predictive performance which is applied to the current predictions.

IV. EXPERIMENTS TO STUDY THE ENSEMBLE PERFORMANCE

This section reports the results of the exploratory experiments that were conducted to study the performance of the proposed framework in terms of the usefulness of the proposed performance measures.

A. Dataset for the Experiments

The KDD Cup 1999 dataset available from the UCI KDD archive [20] was used for the studies. This dataset consist of a wide variety of computer network intrusions (attack types) simulated for a military environment, and is available as two datasets: a training dataset and a test dataset. The 10% version of the training dataset was used for the experiments. This dataset consists of 494,021 instances, 41 features and 23 classes. The 23 classes may be grouped into five categories: NORMAL, DOS, PROBE, R2L and U2R which can then be used as the classes [21]. A new feature (called ID) was added to the dataset with values in the range [1, 494021] as a pseudo timestamp.

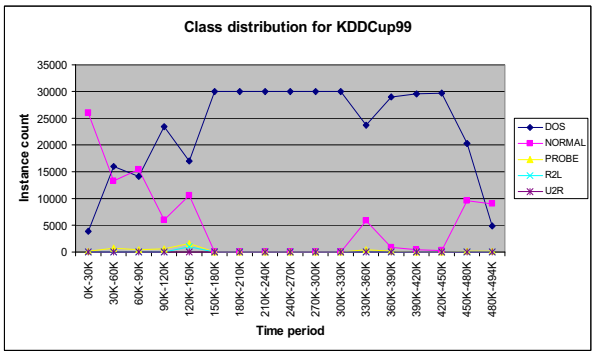

Fig. 3. Class distribution for the KDD Cup 1999 data stream

Fig. 3 shows a plot of the class distribution for this data stream. The data stream also exhibits an extreme imbalance of the class distribution over time. It is clear from Fig. 3 that the classes DOS and NORMAL are the majority classes.

B. Creation of the initial model

The top 90,000 instances of the KDD Cup 1999 dataset were used for the creation of the three base models MW1, MW2 and MW3 for the initial ensemble. The instances were divided into three batches of training instances for each Naïve Bayes base model. Table II shows the instance counts by class for the first three time periods.

TABLE II. CLASS DISTRIBUTION FOR THE TOP 90,000 INSTANCES FOR KDD CUP 1999

class	Class counts for KDD Cup 1999 training data for time period:			Total for class
	0K-30K (W1)	*30K-60K (W2)*	*60K-90K (W3)*	
DOS	3816	15983	14199	33998
NORMAL	26016	13282	15373	54671
PROBE	111	719	391	1221
R2L	53	15	34	102
U2R	4	1	3	8
Total:	30000	30000	30000	90000

C. Experimental Results

Exploratory experiments were conducted to answer the following questions: (1) Does the use of the *Certainty*, *Reliability* and *Incoherence* measures provide useful information for making decisions on model changes? (2) Does the use of instances for which the predicted class belongs to the *certain* or *reliable* category produce base models which provide high predictive performance? The initial ensemble was used to predict the classes of the data stream from time period 90K - 120K to time period 480K - 494K. The prediction performance is shown in Table III. Columns 3, 4, and 5 show the values of the *Certainty*, *Reliability* and *Incoherence* measures which were defined in Section III-B. Column 5 shows the accuracy computed for each time period based on the actual class labels in the dataset. Recall that the *Reliability* measure is used as a surrogate measure of accuracy. A comparison of *Reliability* and *Accuracy* values in Table III indicates that the values are very similar except for one time period (450K - 480K) where there is a large discrepancy.

The initial ensemble was revised by replacing the base models MW1 and MW2 with MW4 and MW5. The new base models were created from data for the time periods 90K - 120K (MW4) and 120K - 150K (MW5). The class labels that were predicted as *certain* or *reliable*, by the initial ensemble were used in the training process. The decision to revise the model was based on the observation that for six consecutive time periods from 150K to 330K, there is only one class (DOS) in the data stream. The revised ensemble model (MW3, MW4, MW5) was used to predict the instances in the time periods from 330K - 360K to 480K - 494K. The training data summary and prediction performance are shown in Table IV and Table V. A

comparison of Tables III and V for the time periods from 330K - 360K to 480K - 494K indicates that the revised ensemble model provides an increase in the *Certainty, Reliability* and *Accuracy* values, and a decrease in the *Incoherence* values. In other words, the model revision leads to improved performance.

TABLE III. PREDICTIVE PERFORMANCE OF INITIAL ENSEMBLE USING MW1, MW2, MW3

Prediction time period	Measures of predictive performance			
	Certainty%	*Reliability%*	*Incoherence%*	*Accuracy%*
90K - 120K	39.3	100	0.0	97.9
120K - 150K	44.4	94.9	5.1	97.0
150K - 330K	0.0	100	0.0	100
330K - 360K	25.9	94.9	5.1	93.0
360K - 390K	62.8	98.5	1.5	96.6
390K - 420K	18.1	100	0.0	100
420K - 450K	0.1	99.7	0.3	99.7
450K - 480K	44.9	94.1	5.9	26.6
480K - 494K	55.2	94.1	5.9	89.8

TABLE IV. CLASS DISTRIBUTION FOR W3, W4, W5 TRAINING INSTANCES

class	Class counts for KDD Cup 1999 training data for time period:			Total for class
	60K-90K (W3)	*90K-120K (W4)*	*120K-150K (W5)*	
DOS	14199	23364	16730	54293
NORMAL	15373	6536	11290	33199
PROBE	391	97	311	799
R2L	34	2	127	163
U2R	3	0	3	6
Total:	30000	29999	28461	88460

TABLE V. PREDICTIVE PERFORMANCE OF REVISED ENSEMBLE USING MW3, MW4, MW5

Prediction time period	Measures of predictive performance			
	Certainty%	*Reliability%*	*Incoherence%*	*Accuracy%*
150K - 330K	0.0	100	0.0	100
330K - 360K	47.2	100	0.0	99.0
360K - 390K	91.3	100	0.0	96.8
390K - 420K	23.3	100	0.0	99.9
420K - 450K	0.4	100	0.0	100
450K - 480K	85.7	100	0.0	32.0
480K - 494K	60.0	100	0.0	95.8

The second ensemble was revised by replacing the base model MW3 with MW12. The MW12 base model was created from data for the time period 330K – 360K. The class labels that were predicted as *certain* or r*eliable*, by the second ensemble were used in the training process. The decision to revise the model was based on the observation that after the six consecutive time periods from 150K to 330K, where only one class (DOS) appears the data stream, the time period 330K – 360K has two classes: DOS and NORMAL. The revised ensemble model (MW4, MW5, MW12) was used to predict the instances in the time periods from 390K - 420K to 480K - 494K. The training data summary and prediction performance are shown in Table VI and Table VII. A comparison of Tables V and VII for the time periods from 390K - 420K to 480K - 494K indicates that the revised ensemble model provides an increase in the *Certainty* values and a decrease in the *Incoherence* values. In other words, the model revision leads to improved levels of confidence in the predictions.

TABLE VI. CLASS DISTRIBUTION FOR THE W4, W5, W12 INSTANCES

class	Class counts for KDD Cup 1999 training data for time period:			Total for class
	90K-120K (W4)	*120K-150K (W5)*	*330K-360K (W12)*	
DOS	23364	16730	23587	63681
NORMAL	6536	11290	6113	23939
PROBE	97	311	298	706
R2L	2	127	0	129
U2R	0	3	2	5
Total:	29999	28461	30000	88460

The experimental results presented in this section lead to the following conclusions. For the KDD Cup 1999 data stream, the selection of training data based on the use of the *certain* and *reliable* prediction categories led to the creation of base models which provided a high level of predictive performance. Secondly, the *Reliability* measure provided a useful surrogate measure of predictive accuracy.

TABLE VII. PREDICTIVE PERFORMANCE OF REVISED ENSEMBLE USING MW4, MW5, MW12

Prediction time period	Measures of predictive performance			
	Certainty%	*Reliability%*	*Incoherence%*	*Accuracy%*
390K - 420K	65.4	100	0.0	99.9
420K - 450K	99.8	100	0.0	100
450K - 480K	99.5	100	0.0	31.9
480K - 494K	98.5	100	0.0	95.2

V. CONCLUSIONS

The purpose of the experimental studies reported in this paper was to assess the usefulness of the proposed measures in supporting the decisions on how to select training data that has been labelled by an ensemble classification model in a data stream mining setting. The first question for the exploratory studies was: Does the use of the *Certainty*, *Reliability* and *Incoherence* measures provide useful information for making decisions on model changes? The second question was: Does the use of instances for which the predicted class belongs to the *certain* or *reliable* category

produce base models which provide high predictive performance? The answer to both questions is yes. The experimental results presented in Section IV have demonstrated that for the KDD Cup 1999 dataset, the Reliability measure provided a good surrogate measure of predictive accuracy for 12 out of 13 (i.e. 92%) of the prediction time periods that were used for the experiments. The experimental results presented in Section IV have also demonstrated that for the KDD Cup 1999 dataset, the use of training data which was labelled and selected based on the prediction categories of *certain* and *reliable* led to the creation of base models which provided a high level of predictive performance for 92% of the prediction time periods. For future work, it will be useful to study how the proposed measures as well as class entropy can be combined to provide a single measure that can be used to determine when model revisions should be performed.

REFERENCES

[1] C.C. Aggarwal, "Data Streams: Models and Algorithms", Kluwer Academic Publishers, Boston, 2007.

[2] J. Gao, W. Fan and J. Han, "On appropriate assumptions to mine data streams: analysis and practice", Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM 2007), IEEE Computer Society, 2007.

[3] M.M. Masud, Q. Chen and J. Gao, "Classification and novel class detection of data streams in a dynamic feature space", Proceedings of European Conference on Machine Learning and Practices in Knowledge Discovery from Databases (ECML/PKDD 2010), LNAI, 337-352, Springer-Verlag, 2010.

[4] J. Gao, W. Fan, J. Han and P.S. Yu, "A general framework for mining concept-drifting data streams with skewed distributions", Proceedings of the SDM Conference, 2007.

[5] G. Hebrial, "Data stream management and mining", In F. Fogelman-Soulié et al. (eds), Mining Massive Data Sets for Security, IOS Press, 2008.

[6] M.M. Gaber, A. Zaslavsky and S. Krishnaswamy, "Mining data streams: a review", SIGMOD Record, vol. 34, no. 2, pp. 18- 26, 2005.

[7] A. Bifet, G. Holmes, R. Kirkby and B. Pfahringer, "Data stream mining: a practical approach", 2011.

[8] T. M. Mitchell, "Machine Learning", Burr Ridge, IL:WCB/McGraw-Hill, 1997.

[9] M.M. Masud, J. Gao, L. Khan and J. Han, "A practical approach to classifying data streams: training with limited amount of data". Proceedings of the 8th IEEE International Conference on Data Mining, pp. 929-934, 2008.

[10] X. Zhu, P. Zhang, X. Lin and Y. Shi, "Active learning from data streams". Proceedings of the 7th IEEE International Conference on Data Mining, pp. 757-762, 2007.

[11] P. Zhang, X. Zhu and L. Guo, "Mining data streams with labelled and unlabelled training examples". Proceedings of the 9th IEEE International Conference on Data Mining, pp. 627-636, 2009.

[12] B. Settles, "Active learning literature survey", Department of Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2009. Available at: http://active-learning.net/

[13] H. Wang, W. Fan, P.S. Yu and J. Han, "Mining concept-drifting data streams using ensemble classifiers". Proceedings of the ACM Special Interests Group on Knowledge Discovery in databases,SIGKDD'2003, pp.226-235, Washington DC, 2003.

[14] J.Z. Kolter and M.A. Maloof, "Dynamic weighted majority: an ensemble method for drifting concepts", Journal of Machine Learning Research, vol. 8, pp. 2755-2790, 2007.

[15] J. Gama, P. Medas and R. Rocha, "Forest trees for on-line data", Proceedings of the ACM Symposium on Applied Computing", pp. 632-636, Cyprus, 2004.

[16] S. Hashemi, Y.Yang, Z. Mirzamomen and M. Kangavari, "Adapted one-versus-all decision trees for data stream classification", IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 5, 2009.

[17] W.N. Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification". Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 377-382. ACM Press, New York, 2001.

[18] R. Munro and S.Chawla, "An integrated approach to mining data streams", Technical Report TR-548, School of Information Technologies, University of Sydney, 2004.

[19] P.E.N. Lutu, "Fast feature selection for Naïve Bayes classification in data stream mining", in S.I. Ao, L. Gelman, D.W.L. Hukins, A. Hunter and A.M. Korsunsky (eds.) Proceedings of the World Congress on Engineering (WCE 2013), London, U.K. July 2013, vol. III, 1549-1554, 2013.

[20] S.D. Bay, D. Kibler, M.J. Pazzani and P. Smyth, "The UCI KDD archive of large data sets for data mining research and experimentation", ACM SIGKDD, vol. 2, no. 2, pp. 81-85, 2000.

[21] S. W. Shin and C. H. Lee, "Using attack-specific feature subsets for network intrusion detection". In Proceedings of the 19th Australian conference on Artificial Intelligence. Hobart, Australia, 2006.