Knowledge and Information Systems manuscript No.
(will be inserted by the editor)

Improved Concept Drift Handling in Surgery Prediction and other Applications

Ayne A. Beyene · Tewelle Welemariam · Marie Persson · Niklas Lavesson

Received: Nov 23, 2012 / Revised: Feb 10, 2014 / Accepted: May 03, 2014

Abstract The article presents a new algorithm for handling concept drift: the Trigger-based Ensemble (TBE) is designed to handle concept drift in surgery prediction but it is shown to perform well for other classification problems as well. At the primary care, queries about the need for surgical treatment are referred to a surgeon specialist. At the secondary care, referrals are reviewed by a team of specialists. The possible outcomes of this review are that the referral: (i) is cancelled, (ii) needs to be complemented, or (iii) is predicted to lead to surgery. In the third case, the referred patient is scheduled for an appointment with a surgeon specialist. This article focuses on the binary prediction of case three (surgery prediction). The guidelines for the referral and the review of the referral are changed due to, e.g., scientific developments and clinical practices. Existing decision support is based on the expert systems approach, which usually requires manual updates when changes in clinical practice occur. In order to automatically revise decision rules, the occurrence of concept drift (CD) must be detected and handled. The existing CD handling techniques are often specialized; it is challenging to develop a more generic technique that performs well regardless of CD type. Experiments are conducted to measure the impact of CD on prediction performance and to reduce CD impact. The experiments evaluate and compare TBE to three existing CD handling methods (AWE, Active Classifier, and Learn++) on one real-world dataset and one artificial dataset. TBA significantly outperforms the other algorithms on both datasets but is less accurate on noisy synthetic variations of the real-world dataset.

Keywords Concept Drift · Surgery Prediction · Trigger-Based Ensemble

A. A. Beyene
E-mail: me.ayne@gmail.com

T. Welemariam
E-mail: tewemit@gmail.com

N. Lavesson and M. Persson
Dept. Computer Science & Engineering, Blekinge Institute of Technology, SE–371 79, Karlskrona, Sweden
E-mail: Niklas.Lavesson@bth.se

1 Introduction

The application of data mining and machine learning techniques has helped many service providers to improve their decision-making process. The healthcare domain applies an increasing number of data mining solutions for decision-making and knowledge discovery. Notable applications include: the analysis of clinical parameters for diagnosis, prediction of the effectiveness of surgical procedures, and discovery of the relationships among clinical and diagnosis data (Magoulas and Prentza, 2001). Today, the demand for non-critical elective surgical care is increasing rapidly. The increase in demand has made patient referral an important process to optimize. Also, surgery is one of the most expensive treatments in the secondary care (Persson and Lavesson, 2009).

One example of the application of supervised learning to improve the referral process is the identification of surgical indicators by mining combined sets of historical patient record data and the corresponding decisions about whether to perform surgery (Persson and Lavesson, 2009). If the need for surgery can be predicted as early as the referral stage, it is possible to optimize the incoming patient queue to the secondary care and the allocation of surgeons and other resources. However, the decision of whether to refer a patient to the secondary care for surgery evolves through time because of changes in scientific developments and clinical practices. Consequently, the performance of the prediction model will decrease because the learned concept becomes invalid due to concept drift (Alippi et al, 2011). The existing decision support systems for patient referral require manual updates when changes in clinical practice occur. Automatically updating the decision support system by handling the concept drift can improve the efficiency of healthcare systems. Accordingly, the impact of concept drift in surgery prediction and the relationship between temporal changes in data distribution and concept drift need to be investigated. Moreover in the state-of-the-art, there are only specific ways of handling concept drift; developing a more generic technique that performs well regardless of concept drift type (e.g.: slow, fast, sudden, gradual, cyclical, noisy) or distribution change is still a challenge (Elwell and Polikar, 2011).

This article presents a concept drift handling algorithm, the Trigger-based Ensemble (TBE), which is based on ensemble-based batch learning (Zhao et al, 2009, Yang et al, 2006) and boosting (Oza and Russell, 2001). TBE predicts the need for surgery without suffering from a significant decrease in prediction performance over time. Experiments are conducted to investigate the impact of concept drift, the relation between concept drift and temporal changes in the data distribution, and to compare the proposed handling algorithm to the state-of-the-art. The algorithms are evaluated on a real-world data set (from the orthopedic department at Blekinge hospital in Sweden) and an artificial data set that represent a different domain. The artifical data set is obtained from the UCI machine learning repository [1]. The artifical data set is included in the experiments to investigate the generalizability of the proposed approach.

The remainder of the article is organized as follows: The next section presents a discussion about surgery prediction and concept drift, which is followed by a review of related work in Section 3. In Section 4, the proposed algorithm is presented. The

[1] The UCI machine learning repository, `http://archive.ics.uci.edu/ml/`

experimental procedure is described in Section 5. Finally, Section 6 concludes the article with an analysis and provides some pointers to future work.

2 Surgery Prediction and Concept Drift

An elective patient referral is a non-emergency case that is commonly submitted by the general practitioner at the primary care. In Sweden, the general practitioner refers a patient when surgery seems necessary. The referral is assessed by a surgeon specialist at the hospital. However, since the general practitioners are not experts on all specialties, unnecessary patient referrals occur quite frequently. Thus, the general practitioners need support on what examinations to conduct before referring a patient.

The elective patient referral contains valuable information that indicates patient surgery need. The indicators can be used as an input to develop an intelligent decision support system for the primary care. The intelligent decision support system predicts patient surgery need and assists the general practitioners in making a correct decision. Hence, unnecessary patient referrals can be reduced through the use of intelligent decision support systems.

In previous work (Persson and Lavesson, 2009), an experiment was conducted in which historical patient records were collected and associated with the corresponding decisions about whether to perform surgery or not. A number of patient record features were identified as surgery indicators. These indicators were used to automatically generate a classification model that could predict the need for surgery as early as during the referral stage on the condition that the general practitioner could run some additional tests during the first examination. The classification model was accurate enough to be used as a basis for optimizing the incoming patient queue to the secondary care. However, the collected data originated from one particular year and it was recognized that the guidelines for taking decisions based on the referral change almost every year.

Thus, regardless of learning algorithm choice, the generated classification model will not always perform well in dynamic environments because the concept acquired during the training phase might have changed (Alippi et al, 2011). The medicine and healthcare management domains are dynamic and complex. Clinical practice, diagnosis procedures, choice of treatment, resource allocation and many other components of healthcare are affected by a vast amount of both known and unknown factors. It is argued that surgery prediction is susceptible to concept drift.

2.1 Concept Drift

Concept drift, in the context of data mining or machine learning, is when prediction models start to perform worse after a period of time. The models lose their performance since the target class or the data distribution of a dataset is changed (Tsymbal, 2004, Wang et al, 2011, Elwell and Polikar, 2011).

There are two types of concept drifts; real and virtual. A real concept drift is a change in the target class which can occur due to changes in a hidden context. A virtual

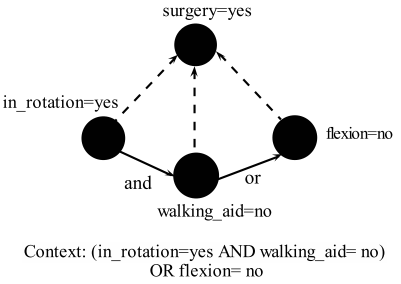
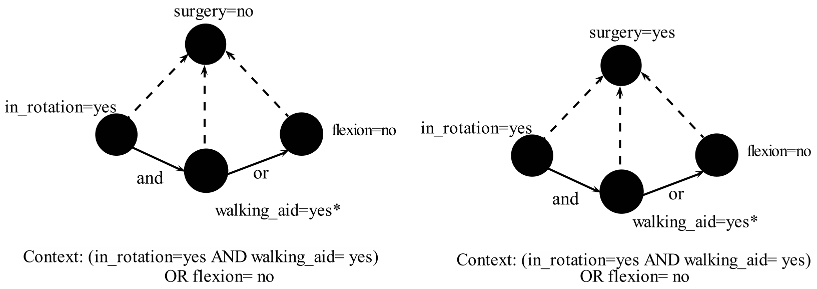


Fig. 1 Original Model Data Observation



(a) Real and Virtual Concept Drift

(b) Virtual Concept Drift

Fig. 2 Concept Drift

concept drift is a change in the data distribution that can occur while the target class remains the same (Tsymbal, 2004, Masud et al, 2010). For real concept drifts, models need to be replaced since the learned concepts become invalid, whereas in virtual concept drift models need additional learning as the error of models may no longer be acceptable. In surgery prediction, concept drift occurs due to scientific development, change in clinical practice, or other changes in data distributions or patterns with unknown causes. Figure 1 shows the concept learned by an algorithm from historical data. The current value of the *walking-aid* attribute is changed to *yes* in both Figure 2(a) and Figure 2(b). Figure 2(a) shows the occurrence of both real and virtual concept drifts, i.e., the target class is changed from *no* to *yes*. In contrast, Figure 2(b) demonstrates that the change does not affect the target class, which indicates virtual concept drift.

There are different types of change in concept drift. The common types are sudden, gradual and recurring. The sudden changes are abrupt when affecting the classification model. For example, when a specific surgery technique is stopped on a legal basis because the treatment is discovered to be hazardous to the surgery outcome and to the patient's health, the concept drift may be sudden. The gradual changes evolve slowly across time, such as when a specific surgery technique is tried out on a specific health situation and proven to be successful by accident, and hence gradually learned by other surgeons. The recurring changes are hidden contexts that reoccur, either cyclically or in an unordered manner. For example, when surgery and drugs are competing treatments for a specific problem and drugs in combination with the surgery technique is gradually changing the concept but cyclically replace each other. A supervised algorithm that learns models for a task for which concept drift occurs needs be able to handle these changes (Widmer and Kubat, 1996).

3 Related Work

STAGGER is the first technique designed to cope with concept drift (Ouyang et al, 2011, Schlimmer and Granger, 1986). It uses a concept description consisting of class nodes connected to attribute-value nodes by probabilistic arcs. The probabilities are updated when new training examples arrive. It also adds nodes corresponding to new classes and new features. To address concept drift, STAGGER decays its probabilities over time. The Active classifier is another single classifier approach that focuses on learning an accurate model with as few labels as possible. It studies how to label selectively instead of asking for all true labels. The method is based on random labeling, a fixed uncertainty strategy, variable allocation of labeling efforts over time, and randomization of the search space. An active classifier encapsulates all the active learning strategies and allows benchmark streaming data experiments through stored, shared, and repeatable settings for synthetic and real-world data (Zliobaite et al, 2011). When a change is detected by the Active Classifier, the old classifier is replaced by a new model. In such a case, recurrent concept drifts may not be handled. The concept drift detection techniques used by the Active Classifier are the Drift detection method (DDM) and Early Drift detection method (EDDM). Both Drift Detection Method(DDM) and Early Drift Detection (EDDM) can be embedded on single classifier as detectors in many concept drift handling techniques (Sobhani and Beigy, 2011).

Gama et al (2004) propose the Drift Detection Method (DDM), each online classifier used to predict the class of an example, that can either true or false. The error rate is modeled by the number of classification errors with a Binomial distribution. However, DDM only detects sudden changes. To improve detection of gradual changes, EDDM extends DDM by relying on the distance between two classification error rates instead of considering only the number of errors (Jose et al, 2006). These concept drift detection methods employ change detection mechanisms, or triggers.

Streaming Ensemble of Algorithms (SEA) is based on a fixed number of ensemble classifiers each constructed from relatively small subsets of data, read sequentially in blocks (Street and Kim, 2001). Once the ensemble is full, new classifiers are added only if they satisfy some quality criterion, based on their estimated ability to improve the ensemble performance. Since the ensemble size is fixed, one of the existing classifiers must be replaced when adding a new model. However, because of the replacement of the ensembles recurrent concepts are not easily addressed. Wang et al (2003) also propose an ensemble of classifiers called the Accuracy Weighted Ensemble. This method maintains classifiers built from batches of training samples, but it weighs each classifier based on their performance on the most recent batch. One of the drawbacks of evolving ensembles of classifiers, in general, is that they build a new base classifier for each batch of new data. Elwell and Polikar (2011) introduce a relatively more generic ensemble of classifiers. It incrementally learns in the presence of concept drift and is commonly called Learn++ or Learn++.NSE. The classifiers are incrementally trained (with no access to previous data) on incoming batches of data, and then combined with weighted majority voting. Classifiers capable of identifying previously unknown instances get more credits while classifiers that misclassify previously known data are penalized. However, since the weight adjustment mechanism used is based on historical classification accuracies, classifiers may get penalized or rewarded wrongly because of noisy input. The inclusion of a noise detector before updating the classifiers would make this framework more effective. In addition, a new set of classifiers is created for each new batch of data. Thus, the ensemble size can become very large over time.

In general, the current approaches are either tested on synthetic data or are studied for specific drift types in specific environments (Elwell and Polikar, 2011). For example, techniques used effectively in spam filtering may not perform well in surgery prediction or weather forecasting. Developing an adaptive learning model that handles concept drift in dynamic environments with the treatment of concept drift and noise is an area of research that demands attention and improvements to the state-of-the-art (Wang et al, 2011, Ouyang et al, 2011).

4 Method

4.1 Trigger Based Ensemble (TBE)

The problem of concept drift in surgery prediction is modeled theoretically to handle changes in concept. The theoretical model combines the trigger and ensemble-based approaches to handle concept drifts. Street and Kim (2001) and Wang et al (2003) argue that an ensemble built by dividing the data into sequential blocks of fixed size examples is effective in handling concept drift. The ensemble-based algorithm handles recurrent concepts by retaining the old concepts in the ensemble (Tsymbal et al, 2006). There are several empirical evaluations that suggest that ensembles perform better than a single classifier (Xiang et al, 2009). However, such ensembles create new classifiers for each block of examples without actively detecting the occurrence of concept drift. This results in an unnecessary growth of the number of classifiers and increase memory consumption. Managing the ensemble size by creating new classifiers only when a concept drift is detected can decrease memory consumption and improve computational efficiency.

The problem of concept drift in surgery prediction is framed by dividing the dataset into sequential blocks of fixed size. Each block, or batch, of patient examples is an m dimensional vector of attributes in some predefined vector space $x = R^m$ and with a class label $y \in \{yes, no\}$. Each batch, b, contains n examples. A sequence of batches is defined as $[(x_i, y_i), (x_{i+1}, y_{i+1}), . ..(x_n, y_n)]$, where the i-th example will be represented by (x_i, y_i). Each incoming patient example is represented as (x, y) in TBE.

4.2 Concept Change Detection

Many parametric approaches assume that the data distribution is Gaussian in nature and can be modeled statistically based on means and covariance. The performance of classification algorithms are commonly evaluated through error and reject rate. Errors are unavoidable due to the existence of uncertainty and noise in classification tasks. Subsequently, a reject rate is introduced as threshold to avoid excessive misclassification (Markou and Singh, 2003). Recently, researchers suggest classification error rate as a concept detection method (Jose et al, 2006, Gama et al, 2004, Nishida and Yamauchi, 2007). Classification error rate-based detection methods are able to detect concept drift from a small number of examples and have less computational costs than other proposed methods (Nishida and Yamauchi, 2007). Thus, classification error rate is selected to monitor concept changes in TBE.

TBE adapts the Early Drift Detection Method (EDDM) to monitor surgery prediction classification error rates. EDDM computes the probability of misclassifying each instance and their standard deviations to monitor change in concept. According to PAC, if the distribution of an example is similar to another example, the classification error rate decreases as the number of examples increases (Mitchell, 1997). With a large number of examples, $n > 30$, the Binomial distribution is closely approximated by a Gaussian distribution with the same mean and variance (Jose et al, 2006). Thus, each batch is chosen so that it includes more than 30 examples in order to make the detection method valid and to monitor classification error rates based on the mean and variance.

4.3 Concept Drift Handling Algorithm

The patient examples arrive in batch, b, over time, t. A knowledge base is initialized by creating a base classifier from the first batch b of data. For each new training dataset, the existence of concept drift is monitored using the EDDM detection method. If a drift is detected, the algorithm adds a new classifier. Otherwise, the classifiers will be trained by updating their weights based on their performance on the current dataset. The weights are increased for classifiers that perform well and vice versa. When the buffer size is full, irrelevant classifiers will be pruned based on error rate and generation time. This pruning helps the model in maintaining the overall competency of the ensemble and preserves memory and computation time for long-term data mining applications (Wang et al, 2003). The final decision of the ensemble is obtained based on majority voting of the current classifiers. This method can be applied with most classification learning algorithms. It can be directly implemented inside online and incremental algorithms, and can be implemented as a wrapper to batch learners. The goal of the proposed method is to detect concept drift from a sequences of examples with a uniform distribution. Those sequences of examples are denoted as context. From the practical point of view, the method tries to select the training set that is the most appropriate to the actual class distribution of the examples.

In general, Trigger Based Ensemble (TBE) is an ensemble algorithm with an embedded active concept drift detector. The active concept drift detector monitors the occurrence of concept change on each incoming batch. A new classifier is added, if and only if a concept drift is detected. This reduces the chance of the ensemble classifiers suffering from outvoting, the growth in the number of incompetent classifiers and memory usage. In TBE, the ensemble handles concept drift by assigning weights to the classifiers based on their classification performance. The final decision is obtained through majority voting. The pseudocode of TBE is presented in Algorithm 1.

5 Experiment

The experiment is conducted using the Massive Online Analysis (MOA) framework. The existing algorithms that are compared with TBE are already available in the MOA framework. The algorithms are selected from different learning paradigms, based on their ability to handle different types of concept drifts. From the single classifier family, the Active Classifier with the early drift detection method is selected. From the ensemble method family, the Accuracy Weighted Ensemble (AWE) and Learn++ are selected. The AWE algorithm is reported to work well on data with reoccurring concepts as well as on different types of drifts. AWE improves its performance gradually over time and is best suited for large data streams. The Active Classifier is a single classifier with concept drift detectors, DDM and EDDM. Learn++ is an incremental ensemble learning algorithm that learns from consecutive batches of data without making any assumptions about the nature or the rate of the drift (Elwell and Polikar, 2011).

The algorithm classification error rate is used as performance evaluation criterion. To make the performance evaluation as fair as possible, similar parameter values are used for all algorithms. For the ensemble and Active Classifier algorithms, the default settings are used to evaluate the classification performance.

5.1 Dataset

There is a shortage of suitable and publicly available real-world benchmark datasets intended for research in data stream classification. Most of the available benchmark datasets are unsuitable for evaluating data stream classification algorithms because the datasets contain too few examples and include insufficient amounts of concept drift. For this reason, it has become common practice to publish results based on both real-world *and* synthetic datasets. The original sample data collected from Blekinge hospital is too limited to derive generalization from. Thus, additional real-world datasets are obtained and synthetic hospital datasets are generated for this purpose. The datasets used are a new hip-replacement dataset from the orthopedics department of Blekinge hospital and the poker-hand dataset from the UCI machine learning repository.

The Hip-replacement dataset includes two years of patient referrals seen at the outpatient clinic of the orthopedics department at Blekinge Hospital. The data are collected for the years 2008 and 2011. Accordingly, a total of 151 complete patient records are identified. Moreover, a total of 80 patient referrals are included from the original hip-replacement dataset from 2008. The identified patient records are preprocessed by

```
Input          : For each dataset $D^t$ where $t = 1, 2, \ldots$
Training Data: $x^t(i) \varepsilon X; y^t(i) \varepsilon Y; i = 1, \ldots, no\_inst$
Supervised learning algorithm: BaseClassifier
for element of b in $D^t$ do
    if $t > 1$ then
        Detect the occurrence of change
        IsConceptChanged $(b)^a$
        for element of i in b do
            $Detected \leftarrow isDetected()$
        end
        if detected then
            if buffer_full then Remove the poorest and oldest classifier, and add a new classifier;
            else add new classifier ;
        end
        if notDetected then
            Compute error of the existing ensemble on new data
            for $i \leftarrow 0$ to $noinst - 1$ do
                $error = \sum 1/noinstance$
            end

        end
        Update and normalize instance weights
        for $i \leftarrow 0$ to $noinst - 1$ do
            if correctlyClassified then $instweight(i) = 1 \div noinstances$
            $total_{weight} + = 1 \div noinstances$
            ;
            else $instweight(i) = (1 \div error) \times (1 \div noinstances)$
            $total_{weight} + = (1 \div error) \times (1 \div noinstances)$
            ;
        end
        for $i \leftarrow 0$ to $noinst - 1$ do
            $instweight(i) = instweight(i)/totalweight$

        end

    else
        Initialize $instweight(i) = 1 \div noinst$
    end
    Call Base Classifier with $D^t$, obtain $h^t : X \rightarrow Y$
    Evaluate all existing classifiers on new data $D^t$
    Compute the weight of each classifier based on its current accuracy on the new data
    Normalize and update the weight of each classifier k
    Obtain the final hypothesis based on majority vote

end
```

[a] Batch

Algorithm 1: TBE Algorithm Pseudocode

removing attributes that could be used to identify specific individuals, irrelevant attributes for the classification and noisy data. Some of the numerical attributes are also discretized in the preprocessing. Finally, a total of 222 existing patient instances are used for experimentation after preprocessing. These instances are defined by 10 input attributes and one binary target attribute.

The hip-replacement dataset is used as a baseline to generate synthetic data. The synthetic data are generated by using two concept generators, STAGGER and SEA, to introduce different types of concept drifts. These concept drift generators are selected based on the types of concept drift they simulate in the real-world dataset.

Both STAGGER and SEA are used to generate a larger amount instances for the hip-replacement dataset from the existing 222 real-world examples. A total of $10,000$ instances are generated for each concept generator. STAGGER creates sequences of data with gradual, abrupt concept drift and noise free examples (Minku et al, 2010, Gama et al, 2004). On the other hand, SEA simulates recurrent and abrupt concept drift to the hip-replacement dataset (Minku et al, 2010). SEA also introduces noise to the dataset it generates. SEA is configured to add 10% noise to the synthetic hip-replacement dataset.

The poker-hand dataset consists of $1,000,000$ instances and 11 attributes. Each instance of the poker-hand dataset is an example of a hand consisting of five playing cards drawn from a standard deck of 52. Each card is described using two attributes, suit and rank, for a total of 10 predictive attributes. There is one class attribute that describes the *poker hand*. The order of cards is important, which is why there are 480 possible Royal Flush hands instead of 4. The poker-hand dataset is used to increase the generalizability of the experimental performance results of the proposed algorithm. The number of instances included in the experiment are limited to $10,000$ because of limitations in computational resources. The first $10,000$ instances that are selected as defining the poker-hand dataset have random cards.

The datasets are divided into chunks, or batches, of data in the experiment. The batch size is determined in proportion to the total number instances. A large batch size results in a stable learner that is suitable for gradual drifts while a small batch size adapts quickly to concept changes, appropriate to abrupt drifts. The Binomial distribution can be approximated by the Gaussian distribution with for a large number of examples, $n > 30$. Accordingly, the batch size is chosen to have 50 to 500 instances so as to make the detection method valid, monitor classification accuracy based on means and covariance and avoid batch related issues. The batch size is set to be 50 and 500 in proportion to the total number of instances.

5.2 Experimental Results and Discussion

Table 1 presents classification error rates before and after handling concept drift. The impact of concept drift in surgery prediction and the relationship between concept drift and temporal changes in the data distribution are shown in Figure 3(a). The relationship between concept drift and temporal changes is presented based on the standard deviation of the error rate between two consecutive batches. The improvement of surgery

Table 1 Performance on the manually modified hip-replacement dataset

Data batch	Instances	Error rate without concept drift handling(SD)[a] Before Handling	Error rate with concept drift handling(SD)[b] After Handling
1	50	0.50(-)	0.30(-)
2	100	0.21(0.21)	0.34(0.23)
3	150	0.16(0.04)	0.22(0.08)
4	200	0.17(0.01)	0.14(0.06)
5[c]	250	0.34(0.12)	0.28(0.10)
6	300	0.54(0.14)	0.56(0.20)
7	350	0.73(0.13)	0.43(0.09)
8	400	0.88(0.11)	0.26(0.12)
9	450	0.90(0.01)	0.19(0.04)
	Mean error rate	**0.49**	**0.30**

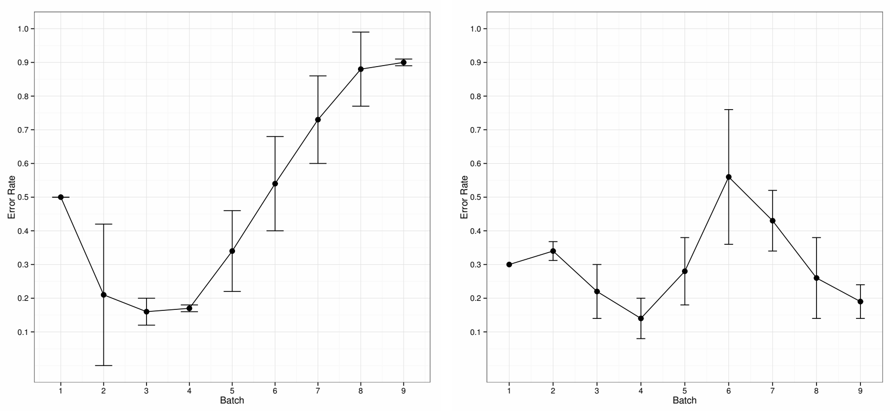
[a] The error rate of each batch before handling concept drift. SD is the standard deviation between two consecutive batches (For instance, between batch 1 and batch 2, between batch 2 and batch 3 and so on.)

[b] The concept drift handling algorithm is TBE

[c] A sudden increase in error rate

prediction performance after handling concept drift is shown in Figure 3(b). Finally, the performance and rank of the four concept drift handling algorithms (Active Classifier, AWE, Learn++, and TBE), are shown in Table 2 through Table 4. Moreover, the performances of the four concept drift handling algorithms are visually presented in Figure 4 through Figure 6. Non-parametric statistical analyses are conducted on the performance results to determine if there are significant performance differences between the compared algorithms.

The experimental results in Table 1 illustrate the classification performance of surgery prediction on the manually modified hip-replacement dataset. The dataset is modified by replicating the real-word hip-replacement dataset and introducing a real concept drift by changing only the class value of the replicated hip-replacement dataset. A total of 444 instances are included in the modified hip-replacement dataset. The classification performance is measured based on error rate. In Column 3 of Table 1, the result of the classification error rate over sequences of batches before handling concept drift is presented. Likewise, Column 4 of Table 1 is the result of the classification error rate over sequences of batches after handling concept drift with TBE. The standard deviation between two consecutive batches indicates the deviation of the current classifier performance from the previous classifier performance. Figure 3(a) depicts the error rate of surgery prediction over sequences of batches before handling concept drift. The figure on the right side, Figure 3(b), depicts the classification error rate of surgery prediction after handling concept drift. The error bars that originate at the classification error rate of each batch represent the deviation in performance between two consecutive batches, b and $b-1$.



(a) Real and Virtual Concept Drift

(b) Virtual Concept Drift

Fig. 3 Hip-replacement dataset prediction performance before and after handling concept drift.

Evaluation of Concept Drift Impact on Surgery Prediction

The result obtained from the experiment in Column 3 of Table 1 is used to evaluate the degree of prediction performance changes over a sequence of batches in the occurrence of concept drift. If the distribution of examples are assumed to be drawn independently from an identical distribution, the classification error rate decreases as the number of examples increases (Mitchell, 1997, McAllester, 1998). Similarly, in Figure 3(a), the classification error rate decreases in the first few batches as the number of instances increase. However, the error rate increases suddenly at the 5-th batch due to changes in the concept. The experimental results are analyzed statistically to evaluate the effect of concept drift in surgery prediction performance. The non-parametric Wilcoxon rank sum test, is used with confidence level 0.05. Accordingly, the test result indicates that classification performance significantly decreases in the occurrence of concept drift. Thus, the null hypothesis, that classification performance is unaffected by the occurrence of concept drift, is rejected for $p < 0.05$.

The relationship between temporal changes in data distribution and surgery prediction is shown based on the standard deviation of the classification error rate of two consecutive batches. As discussed in Section 4.3, a sequence of instances are treated in batches that arrive at different points in time. The standard deviation between the classification error rates of two consecutive batches is depicted in Figure 3(a) and Figure 3(b), with error bars. A variance in prediction performance indicates concept change. The Levene test is used to check the probability of batches being drawn from the same distribution. Accordingly, the test result indicates that the variance of classification performance differs between batches (for $p < 0.05$). Hence, significant temporal changes in the data distribution indicate concept drift. TBE adapts the EDDM detection method that monitors the occurrence of concept drift by computing the standard deviation of misclassified instances.

Performance Evaluation Before and After Handling Concept Drift

The negative impact of concept drift in surgery prediction is reduced through a concept drift handling algorithm as can be viewed in Figure 3(b). The experimental results in Column 4 of Table 1 are statistically analyzed using the Wilcoxon paired-pairs signed-ranks test with confidence level 0.05. The test result shows that concept drift handling improves classification performance significantly. That is, the null hypothesis is rejected with $p < 0.05$. Similarly, the plot in Figure 3(b), reveals a decrease in the classification error rate after handling the occurrence of concept drift, at the 6-th batch; whereas Figure 3(a) describes how the error rate increases due to the occurrence of concept drift. This shows that concept drift handling improves classification performance in general. However, the amount of improvement in prediction performance depends on the type of concept drift handling method used. It is important to recognize that it is difficult to define a generally acceptable error rate for the surgery prediction task. The acceptable level of performance is related to the reason for conducting the prediction task (for example, to improve resource allocation or to decrease waiting times for a specific group of patients).

In the remaining experiments, the plausible combinations of the existing detection and handling algorithms including TBE are evaluated on a synthetic hip-replacement dataset generated by STAGGER and SEA. The experiments compare the performance of handling concept drift of four handling algorithms (AWE, Active Classifier, Learn++, TBE). The four algorithms are statistically analyzed using Friedman's two-way analysis of variance in conjunction with the Nemenyi post-hoc test. The Friedman test is used to determine whether there is any significant differences in performance between the algorithms. If a significant difference is detected the Nemenyi post-hoc test is used to conduct pair-wise comparisons of the performance of the algorithms, to determine which algorithms differ significantly in performance and in which direction. The experimental results are illustrated in Table 2 through Table 4 and in Figure 4 through Figure 6, respectively.

From Figure 4 to Figure 6 the performances of AWE, Active Classifier, Learn++, and TBE on the real-world dataset and the artificial Poker dataset are presented. Figure 4 depicts the decreasing error rates of the four algorithms (AWE, Active Classifier, Learn++, and TBE) on the synthetic hip-replacement dataset generated by the STAGGER concept generator. Similarly, Table 2 illustrates the classification accuracies and corresponding ranks of the four compared algorithms on the same dataset.

Performance Evaluation on the STAGGER-based Hip-replacement Dataset

The four included algorithms are compared on a synthetic dataset: the hip-replacement dataset is used as a basis by the STAGGER concept generator to produce a synthetic variant of the dataset with $10,000$ instances. As in earlier comparisons of the algorithms, Friedman's test is used to statistically analyze the performance results to determine whether there are significant differences. Again, the Nemenyi post-hoc test is used for pair-wise comparisons if a significant difference is detected. A confidence level of 0.05 is used.

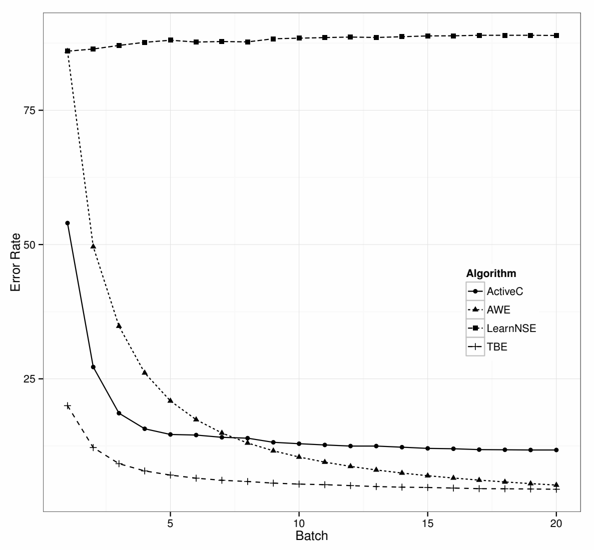


Fig. 4 Performance Comparison on Handling CD (STAGGER Hip-replacement Dataset)

Table 2 Comparison of the accuracy(%) of the included concept drift handling algorithms on the hip-replacement data set simulated by STAGGER concept generator

Batch	Instances	TBE(Rank[a])	AWE(Rank)	Learn++(Rank)	Active Classifier(Rank)
1	500	**80.00**(1)	14.00(3.5)	14.00(3.5)	46.00(2)
2	1,000	**87.80**(1)	50.40(3)	13.60(4)	72.80(2)
3	1,500	**90.80**(1)	65.20(3)	12.93(4)	81.40(2)
4	2,000	**92.15**(1)	73.90(3)	12.35(4)	84.30(2)
5	2,500	**92.92**(1)	79.12(3)	11.96(4)	85.36(2)
6	3,000	**93.50**(1)	82.60(3)	12.30(4)	85.47(2)
7	3,500	**93.89**(1)	85.09(3)	12.20(4)	85.89(2)
8	4,000	**94.13**(1)	86.95(2)	12.28(4)	86.05(3)
9	4,500	**94.42**(1)	88.40(2)	11.69(4)	86.82(3)
10	5,000	**94.60**(1)	89.56(2)	11.58(4)	87.08(3)
11	5,500	**94.71**(1)	90.51(2)	11.47(4)	87.31(3)
12	6,000	**94.88**(1)	91.30(2)	11.37(4)	87.52(3)
13	6,500	**95.06**(1)	91.97(2)	11.45(4)	87.52(3)
14	7,000	**95.17**(1)	92.54(2)	11.31(4)	87.73(3)
15	7,500	**95.23**(1)	93.04(2)	11.16(4)	87.95(3)
16	8,000	**95.34**(1)	93.48(2)	11.14(4)	88.03(3)
17	8,500	**95.45**(1)	93.86(2)	11.04(4)	88.18(3)
18	9,000	**95.47**(1)	94.20(2)	11.04(4)	88.21(3)
19	9,500	**95.51**(1)	94.51(2)	11.04(4)	88.25(3)
20	10,000	**95.56**(1)	94.78(2)	11.08(4)	88.25(3)
Average Rank		**1**	**3.02**	**3.98**	**2.65**

[a] The rank is computed based on accuracy

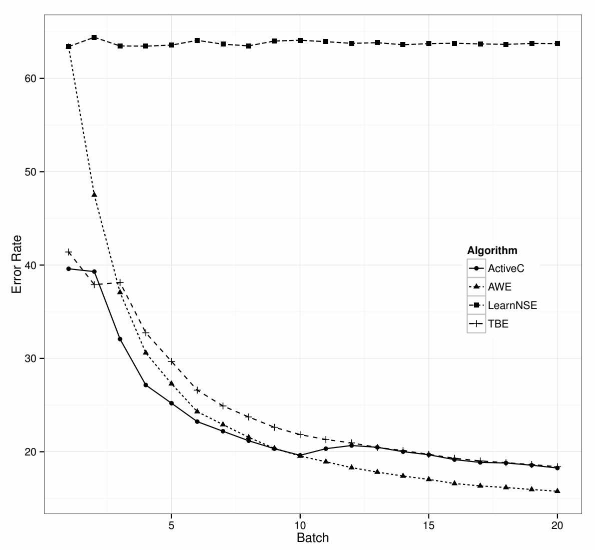


Fig. 5 Performance Comparison on Handling CD (SEA Hip-replacement Dataset)

Table 2 illustrates the classification performance of the four algorithms (AWE, Learn++, Active Classifier, and TBE). The performance of each algorithm is ranked. The average ranks provide a reasonable comparison of the algorithms (Demsar, 2006). The Friedman test checks whether the average ranks are significantly different from the mean rank. The average rank is computed based on each algorithm's ranks. Consequently, the test result indicates a significant difference between the four algorithms (with $p < 0.05$). Since the null-hypothesis is rejected, further analyses are conducted with the Nemenyi post-hoc test to compare the algorithms to each other. The performance of two classifiers is significantly different if the corresponding average ranks differ by at least the critical difference. Consequently, the accuracy of TBE is significantly higher than those of the remaining algorithms (with $p < 0.05$).

In addition to the statistical test results, Figure 4 also provides visual information that TBE has lower error rate starting from the first batch compared to the other algorithms. This indicates that TBE handles concept drift better than the other algorithms with a low variation throughout the batches. TBE also performs in a consistent manner by handling the occurrence of concept drifts for both small and large datasets. Overall, the performance of TBE is significantly better than AWE, Active Classifier and Learn++.

Figure 5 shows the performances of the four algorithms, AWE, Active Classifier, Learn++ and TBE, on the $10,000$-instance synthetic hip-replacement dataset generated by the SEA concept generator. SEA simulates recurrent, abrupt drifts and adds 10% of noisy data. The figure shows the error rate of the concept drift handling algorithms over sequences of batches.

Table 3 Comparison of accuracy(%) between concept drift handling algorithms on the hip-replacement data set generated by the SEA concept generator

Batch	Instances	TBE(Rank[a])	AWE(Rank)	Learn++(Rank)	Active Classifier(Rank)
1	5,00	58.60(2)	36.60(3.5)	36.60(3.5)	**60.40**(1)
2	1,000	**62.10**(1)	52.50(3)	35.60(4)	60.70(2)
3	1,500	61.87(3)	62.93(2)	36.53(4)	**67.93**(1)
4	2,000	67.25(3)	69.40(2)	36.55(4)	**72.85**(1)
5	2,500	70.32(3)	72.72(2)	36.44(4)	**74.80**(1)
6	3,000	73.40(3)	75.70(2)	35.93(4)	**76.77**(1)
7	3,500	75.09(3)	77.09(2)	36.34(4)	**77.80**(1)
8	4,000	76.28(3)	78.45(2)	36.53(4)	**78.83**(1)
9	4,500	77.38(3)	79.64(2)	36.00(4)	**79.67**(1)
10	5,000	78.16(3)	**80.46**(1)	35.92(4)	80.38(2[b])
11	5,500	78.69(3)	**81.07**(1)	36.07(4)	79.67(2)
12	6,000	79.05(3)	**81.70**(1)	36.25(4)	79.33(2)
13	6,500	79.54(2)	**82.18**(1)	36.18(4)	79.52(3)
14	7,000	79.87(3)	**82.60**(1)	36.40(4)	79.99(2)
15	7,500	80.28(3)	**82.96**(1)	36.28(4)	80.32(2)
16	8,000	80.70(3)	**83.40**(1)	36.24(4)	80.84(2)
17	8,500	80.98(3)	**83.66**(1)	36.31(4)	81.14(2)
18	9,000	81.17(3)	**83.83**(1)	36.36(4)	81.20(2)
19	9,500	81.37(3)	**84.04**(1)	36.26(4)	81.44(2)
20	10,000	81.60(3)	**84.23**(1)	36.28(4)	81.74(2)
Average Rank[c]		**2.80**	**1.58**	**3.98**	**1.65**

[a] The rank is assigned based classification accuracy

[b] Accuracy starts to decrease

[c] The average rank of the algorithms.

Figure 4 and Figure 5 indicates that LearnNSE maintains a high error rate across batches while the remaining algorithms manage to reduce the error rate as new batches are processed. The reductions seem to exhibit exponential behavior during the first ten batches. Table 3 illustrates the classification accuracies and ranks of the four algorithms on the hip-replacement data set generated by the SEA concept generator.

Performance Evaluation on SEA-based Hip-replacement Dataset

The performances of the four algorithms on the synthetic dataset generated from the hip-replacement dataset by the SEA concept generator are statistically analyzed using the Friedman test. A confidence level of 0.05 is employed. If significant differences are detected, further analysis is conducted with the Nemenyi post-hoc test to find out which algorithm performs better than other algorithms.

Table 3 illustrates the comparison between the four algorithms (AWE, Learn++, Active Classifier, and TBE). The average ranks provide a fair comparison of the algorithms. The Friedman test checks whether the average ranks are significantly different from the mean rank, 2.5. The average rank is computed based on accuracy of the algorithms. Consequently, the test result indicates a significant difference between the four

Table 4 Comparison of the accuracy(%) of the concept drift handling algorithms on the Poker dataset

Batch	Instances	TBE(Rank[a])	AWE(Rank)	Learn++(Rank)	Active Classifier(Rank)
1	500	32.40(2.5)	32.40(2.5)	32.40(2.5)	**38.00**(1)
2	1,000	**52.80**(1.5)	51.50(4)	**52.80**(1.5)	52.10(3)
3	1,500	**56.26**(1)	48.19(4)	51.06(3)	52.86(2)
4	2,000	**62.00**(1)	46.45(4)	49.05(3)	50.55(2)
5	2,500	**65.64**(1)	41.76(4)	55.27(3)	56.59(2)
6	3,000	**65.26**(1)	43.46(4)	53.63(3)	59.66(2)
7	3,500	**68.25**(1)	42.17(4)	54.17(3)	58.68(2)
8	4,000	**70.35**(1)	43.10(4)	58.02(3)	61.30(2)
9	4,500	**72.00**(1)	44.24(4)	54.95(3)	63.62(2)
10	5,000	**74.02**(1)	41.12(4)	58.06(3)	64.74(2)
11	5,500	**74.61**(1)	40.80(4)	60.10(3)	65.52(2)
12	6,000	**74.35**(1)	39.46(4)	61.05(3)	65.68(2)
13	6,500	**73.87**(1)	40.98(4)	60.06(3)	66.01(2)
14	7,000	**72.64**(1)	40.34(4)	60.00(3)	65.85(2)
15	7,500	**72.65**(1)	41.08(4)	60.68(3)	66.25(2)
16	8,000	**72.33**(1)	41.80(4)	59.47(3)	67.41(2)
17	8,500	**72.81**(1)	43.77(4)	60.50(3)	67.80(2)
18	9,000	**73.38**(1)	44.33(4)	59.64(3)	67.43(2)
19	9,500	**73.64**(1)	45.23(4)	60.66(3)	68.09(2)
20	10,000	**73.74**(1)	43.84(4)	60.95(3)	67.64(2)
Average Rank		**1.10**	**3.93**	**2.90**	**2.00**

[a] The rank is assigned based on accuracy.

algorithms (with $p < 0.05$). A further post-hoc test is conducted for pairwise comparisons. Accordingly, the error rate of AWE and Active classifier is significantly lower than TBE and Learn++ (with $p < 0.05$). The error rate of AWE is insignificantly different to Active Classifier. Overall, AWE and Active classifier are significantly better than the other algorithms and TBE is found to be relatively less capable next to Learn++ in handling noisy data.

In addition to the statistical test result, Figure 5 provides visual information about the four algorithms. AWE starts with a very high error rate but outperforms the other algorithms by adapting quickly to changes as the number of examples increases. Active classifier starts with a better performance than the other algorithms but the error rate increases starting from the 10-th batch. TBE starts with better performance next to the Active classifier but adapts gradually for both small and large datasets by maintaining the performance of existing classifiers. Table 4 illustrates the classification accuracy of

the four algorithms on the Poker dataset. The performance of each algorithm is measured using classification accuracy. Similarly, Figure 6 depicts the performances of the four algorithms, AWE, Active Classifier, Learn++ and TBE, on the $10,000$-instance Poker dataset.

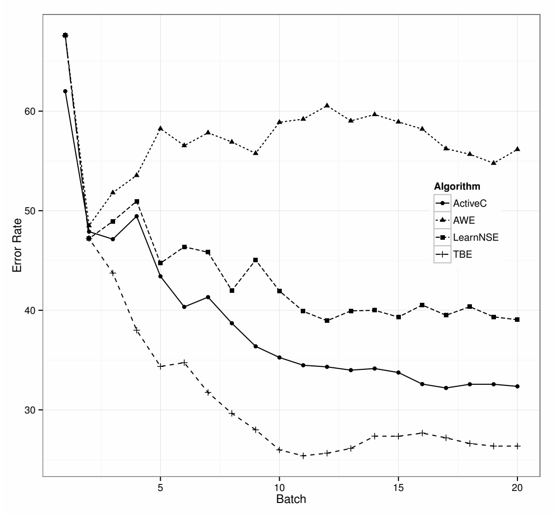


Fig. 6 Performance comparison on Handling CD (Poker dataset)

Performance Evaluation on the Poker Dataset

For generalization purposes, the four algorithms (AWE, Active Classifier, Learn++, and TBE) are also evaluated on another domain using the Poker dataset. This part of the experiment is unrelated to surgery prediction. Instead, the aim is to investigate whether the results achieved on the surgery prediction case translate over to other domains. The statistical testing procedure is identical to the earlier experimental comparisons of the four algorithms for the surgery prediction case.

Table 4 illustrates a fair comparison of the algorithms (AWE, Learn++, Active Classifier, and TBE) based on average ranks. The Friedman test is again used to determine whether the average ranks are significantly different from the mean rank, 2.48. The average rank is computed based on the ranks of the classification accuracies. As a consequence, there is a significant difference between the four algorithms (with $p < 0.05$). Further analysis is conducted with the Nemenyi post-hoc test for pairwise comparisons. Accordingly, the error rate of TBE is significantly lower than those of AWE and Learn++ (with $p < 0.05$). On the other hand, the error rate of Active Classifier is insignificantly different from TBE and Learn++ but significantly lower than AWE.

Experimental Summary

To summarize, the experimental results from Figure 4 to Figure 6 illustrate that TBE performs better on average compared to the other algorithms. TBE performed significantly better than the other algorithms on the hip-replacement data set simulated by STAGGER and on the Poker dataset. Moreover, the results show that TBE handles concept drift in a consistent manner for both small and large datasets. However, AWE and Active Classifier perform better on a noisy hip-replacement dataset that is simulated by the SEA concept generator.

Future Work

The future research has three main directions: (i) optimizing the performance of TBE and developing noise handling capabilities, (ii) performing additional experiments on datasets from other domains with different characteristics to validate and optimize the performance of the Trigger-based Ensemble method, and (iii) investigating how to properly distinguish real concept drift from noise for selected domains. The additional experiments should also include additional performance measures as well as time, complexity, and memory consumption measurements. In this article, we have used error rate as the primary performance measure. The rationale for this decision is that error rate is the predominantly employed measure in comparable studies on concept drift. The authors believe, however, that a more deepened analysis of the performance impact of concept drift is warranted. Subsequent experiments should therefore encompass alternative performance measures such as the area under the precision–recall curve, sensitivity, and specificity. In addition, the computational complexity of concept drift detection and handling is an important aspect in this area of research. A simplified complexity analysis can of course be performed by reviewing the presented algorithm pseudocode. We are, however, planning a future theoretical analysis of the TBE detection and handling algorithms.

6 Conclusion

This article investigated a concept drift handling algorithm designed for the surgery prediction task, which is a supervised learning task where the aim is to generate a mapping between patient record instances and a binary target indicating whether the corresponding patient needs surgery or not. Today, this task is performed manually. The general practitioner who examines the patient need to decide whether to refer the patient to the secondary care for specialist treatment. The scope is delimited to elective surgical specialist care. If the surgery prediction task is to be automated, it is important to be able to handle concept drift since the guidelines for making decisions about the need for surgery are revised, sometimes gradually and sometimes abruptly.

The occurrence of concept drift in the hip-replacement dataset caused a sudden decrease in classification performance. If the distribution from which examples are drawn is similar or identical across batches, the classification performance should not decrease for subsequent (and larger) batches. However, the results on the hip-replacement dataset indicate that the error rate increases as the number of examples increase. There is a significant variation in classification performance, when the learned concept is changed. The negative effect of concept drift can be reduced through concept drift handling algorithms.

State-of-the-art concept drift handling algorithms (based on either single classifiers or ensemble approaches) are here evaluated on a real-world dataset of patients with hip problems and the Poker hand dataset from the UCI machine learning repository (the latter is used for generalization purposes). The investigation led to the development of an algorithm called the Trigger-based Ensemble, which is based on ensemble learning and boosting. This proposed algorithm showed a comparatively better ability to detect and handle concept drift than the state-of-the-art algorithms.

The Trigger-based Ensemble actively detects the occurrence of changes on each incoming batch of instances and adapts to the changes incrementally. It uses the current and past predictions of classifiers combined with dynamically updated voting weights. It is assumed that adding an active detector reduces the chance of ensemble classifiers suffering from outvoting, that is, the growth of the number of incompetent classifiers. Moreover, the ensemble size does not become overly large. Thus the contribution of this research is twofold; improving the performance on the surgery prediction task and presenting a generic concept drift handling algorithm that performed comparatively better than the existing concept drift handling algorithms.

Acknowledgements The conducted research is funded by a grant from Blekinge county research council. The authors would like to thank Elisabeth Håkansson at Blekinge hospital for extracting and compiling patient records as well as the corresponding decisions concerning surgery to enable the generation of the real-world hip-replacement dataset.

Bibliography

Alippi C, Boracchi G, Roveri M (2011) An effective just-in-time adaptive classifier for gradual concept drifts. International Joint Conference on Neural Networks pp 1675–1682

Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. Machine Learning Research 7:1–30

Elwell R, Polikar R (2011) Incremental learning of concept drift in nonstationary environments. IEEE Transactions on Neural Networks 22(10):1517–1531

Gama J, Medas P, Castillo G, Rodrigues P (2004) Learning with drift detection. In: Advances in Artificial Intelligence, Springer, pp 286–295

Jose MBG, Campo-Avila JD, Fidalgo R, Bifet A, Gavalda R, Morales-bueno R (2006) Early drift detection method. In: ECML/PKDD Workshop on Knowledge Discovery from Data Streams, ACM Press, pp 77–86

Magoulas GD, Prentza A (2001) Machine learning in medical applications. In: Machine Learning and Its Applications, Advanced Lectures, Springer, pp 300–307

Markou M, Singh S (2003) Novelty detection: A review - part 1: Statistical approaches. Signal Processing 83

Masud MM, Chen Q, Khan L, Aggarwal C, Gao J, Han J, Thuraisingham B (2010) Addressing concept-evolution in concept-drifting data streams. In: IEEE International Conference on Data Mining, IEEE Press, pp 929–934

McAllester DA (1998) Some PAC-Bayesian theorems. In: Annual Conference on Computational Learning Theory, ACM Press, pp 230–234

Minku L, White A, Yao X (2010) The impact of diversity on online ensemble learning in the presence of concept drift. IEEE Transactions on Knowledge and Data Engineering 22(5):730–742

Mitchell TM (1997) Machine Learning. McGraw-Hill

Nishida K, Yamauchi K (2007) Detecting concept drift using statistical testing. In: International Conference on Discovery Science, Springer, pp 264–269

Ouyang Z, Zhao Z, Gao Y, Wang T (2011) Study on the classification of data streams with concept drift. International Conference on Fuzzy Systems and Knowledge Discovery pp 1673–1677

Oza NC, Russell S (2001) Experimental comparisons of online and batch versions of bagging and boosting. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, pp 359–364

Persson M, Lavesson N (2009) Identification of surgery indicators by mining hospital data: A preliminary study. In: International Conference on Database and Expert Systems Application, pp 323–327

Schlimmer J, Granger R (1986) Beyond incremental processing: Tracking concept drift. In: National Conference on Artificial Intelligence, Morgan Kaufmann, pp 502–507

Sobhani P, Beigy H (2011) New drift detection method for data streams. In: International Conference on Adaptive and Intelligent Systems, Springer, pp 88–97

Street WN, Kim Y (2001) A streaming ensemble algorithm (sea) for large-scale classification. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, pp 377–382

Tsymbal A (2004) The problem of concept drift: Definitions and related work. Computer Science Department, Trinity College Dublin

Tsymbal A, Pechenizkiy M, Cunningham P, Puuronen S (2006) Handling local concept drift with dynamic integration of classifiers: Domain of antibiotic resistance in nosocomial infections. In: IEEE Symposium on Computer-Based Medical Systems, IEEE Computer Society, pp 679–684

Wang H, Fan W, Yu PS, Han J (2003) Mining concept-drifting data streams using ensemble classifiers. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, pp 226–235

Wang S, Schlobach S, Klein M (2011) Concept drift and how to identify it. Web Semantics Science Services and Agents on the World Wide Web 9(3):247–265

Widmer G, Kubat M (1996) Learning in the presence of concept drift and hidden contexts. Machine Learning 23:69–101

Xiang C, Chen M, Wang H (2009) An ensemble method for medicine best selling prediction. In: International Conference on Fuzzy Systems and Knowledge Discovery, vol 1, pp 100–103

Yang Y, Wu X, Zhu X (2006) Mining in anticipation for concept change: Proactive-reactive prediction in data streams. Data Mining and Knowledge Discovery 13(3):261–289

Zhao QL, Jiang YH, Xu M (2009) A fast ensemble pruning algorithm based on pattern mining process. Data Mining and Knowledge Discovery 19(2):277–292

Zliobaite I, Bifet A, Pfahringer B, Holmes G (2011) Active learning with evolving streaming data. In: European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, pp 597–612

Author Biographies

Ayne A. Beyene received her B. Sc. in management information systems and M. Sc. in computer science from Unity University, Ethiopia and Blekinge Institute of Technology, respectively. Currently, she is working as a software developer consultant at Ants. She has a great interest in the research area of Artifical intelligence, mainly in data mining and machine learning.

Tewelle Welemariam received his B.Sc. in Computer Science and Engineering from Mekelle Institute of Technology, Ethiopia and his M.Sc. in Computer Science from Blekinge Institute of Technology, Sweden in 2007 and 2012, respectively. Tewelle has several years of industrial experience as a Software Engineer and Researcher. He is interested in solving real-world industrial challenges. He currently works as a software engineer at International Copyright Enterprise (ICE) in Stockholm, Sweden.

Dr. Marie Persson defended her Ph.D. in Computer Science in May 2010. Her main interests are in Health Care Logistics, that is, how concepts from logistics and optimization can be applied in Health Care Delivery to improve performance and help management decisions. She is a registered intensive care nurse and has a Master of Science degree in Software Engineering from Blekinge Institute of Technology, 2004.

Dr. Niklas Lavesson is Associate Professor of Computer Science at Blekinge Institute of Technology in Karlskrona, Sweden. He received his M.Sc. degree in Software Engineering and Ph.D. degree in Computer Science, in 2003 and December 2008 respectively, from Blekinge Institute of Technology. His main area of research is Machine Learning with a special focus on analysis and evaluation of supervised learning algorithms. The applied research interests include data mining and knowledge discovery in healthcare management and operations management.