IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

Study of Detecting and Localizing Concept Drifts from Event Logs in Process Mining

Trupti Kakkad[1], Dr. Rahila Sheikh[2]

Rajiv Gandhi College of Engineering and Research Technology Chandrapur, Babupeth, Chandrapur [1, 2]

Abstract: Most business processes change over time, contemporary process mining techniques tend to analyse these processes as if they are in a steady state. Processes may change suddenly or gradually. The drift may be periodic (e.g., because of seasonal influences) or one-of-a-kind (e.g., the effects of new legislation). For the process management, it is crucial to discover and understand such concept drifts in processes. A generic framework and specific techniques to detect when a process changes and to localize the parts of the process that have changed. Different features are proposed to characterize relationships among activities. These features are used to discover differences between successive populations. The approach has been implemented as a plug-in of the ProM process mining framework and has been evaluated using both simulated event data exhibiting controlled concept drifts and real-life event data from a Dutch municipality.

Keywords: Concept drift, flexibility, hypothesis tests, process changes, process mining.

I. INTRODUCTION

Business processes are nothing more than logically related tasks that use the resources of an organization to achieve a defined business outcome. Business processes can be viewed from a number of perspectives, including the control flow, data, and the resource perspectives. In today's dynamic marketplace, it is increasingly necessary for enterprises to streamline their processes so as to reduce cost and to improve performance.

In addition, today's customers expect organizations to be flexible and adapt to changing circumstances. New legislations such as the WABO act [1] and the Sarbanes–Oxley Act [2], extreme variations in supply and demand, seasonal effects, natural calamities and disasters, deadline escalations [3], and so on, are also forcing organizations to change their processes. For example, governmental and insurance organizations reduce the fraction of cases being checked when there is too much of work in the pipeline Another example, in a disaster, hospitals, and banks change their operating procedures. It is evident that the economic success of an organization is more and more dependent on its ability to react and adapt to changes in its operating environment.

Therefore, flexibility and change have been studied in-depth in the context of business process management (BPM). For example, process-aware information systems (PAISs) [4] has been extended to be able to flexibly adapt to changes in the process. State-of-the-art workflow management (WFM) and BPM systems [5] provide such flexibility, e.g., we can easily release a new version of a process. In addition, in processes not driven by WFM/BPM systems (such as the usage of medical systems) there is even more flexibility as processes are controlled by people rather than information systems.

Many of today's information systems are recording an abundance of event logs. Process mining is a relatively young research discipline aimed at discovering, monitoring, and improving real processes by extracting knowledge from event logs

There is a need for techniques that deal with such second-order dynamics. Analysing such changes is of utmost importance when supporting or improving operational processes and to obtain an accurate insight on process executions at any instant of time. When dealing with concept drifts in process mining, the following three main challenges emerge.

1) Change point detection: The first and most fundamental problem is to detect concept drift in processes, i.e., to detect that a process change has taken place.
2) Change localization and characterization: Once a point of change has been identified, the next step is to characterize the nature of change, and identify the region(s) of change (localization) in a process.
3) Change process discovery: Having identified, localized, and characterized the changes, it is necessary to put all of these in perspective. There is a need for techniques/tools that exploit and relate these discoveries.

There are mainly two challenges: 1) change (point) detection and change localization and 2) characterization in an offline setting (Fig. 1).

They define different features and propose a framework for dealing with these two problems from a control-flow perspective. Initially, they show the promise of the techniques on a synthetic log and later evaluate them on a real-life case study from a large Dutch municipality.

DOI 10.17148/IJARCCE.2016.5969

II. LITERATURE SURVEY

In this section, the basic concepts in process mining and concept drifts in data mining/machine learning.

A. Process Mining:

Process mining serves a bridge between data mining and business process modelling [6]. Business processes leave trails in a variety of data sources (e.g., audit trails, databases, and transaction logs). Process mining aims at discovering, monitoring, and improving real processes by extracting knowledge from event logs recorded by a variety of systems (ranging from sensor networks to enterprise information systems). The starting point for process mining is an event log, which is a collection of events. That events can be related to process instances (often called cases) and are described by some activity name. The events within a process instance are ordered. Therefore, a process instance is often represented as a trace over a set of activities. In addition, events can have attributes such as timestamps, associated resources (e.g., the person executing the activity), transactional information (e.g., start, complete, suspend, and so on), and data attributes (e.g., amount or type of customer). For a more formal definition of event logs used in process mining, the reader is referred to [6]. Fig. 2 shows a fragment of an example log. Event logs like in Fig. 2 are completely standard in the process mining community and event log formats such as MXML [7] and XES [8] are used. The topics in process mining can be broadly classified into three categories:

1) Discovery;
2) Conformance; and
3) Enhancement.

Process discovery deals with the discovery of models from event logs. These models may describe control flow, organizational aspects, time aspects, and so on. For example, there are dozens of techniques that automatically construct process models (e.g., Petri nets or BPMN models) from event logs [6].

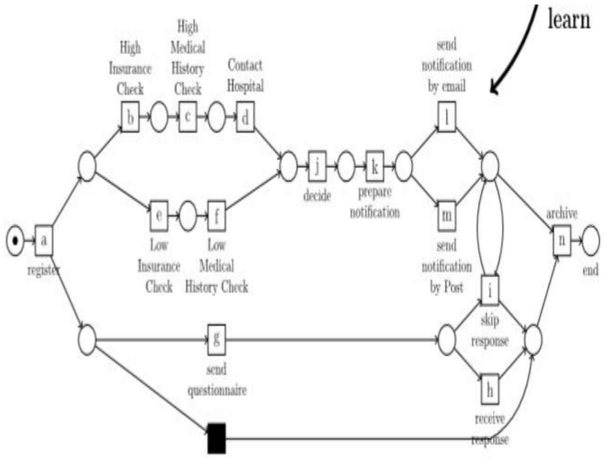


Fig 1: Process Delivery.

Fig. 1 shows the basic idea of process discovery. An event log containing detailed information about events is transformed into a multiset of traces L = [abcdjkln, aefjkmn, abgchdjkln,]. Process discovery techniques are able to discover process models such as the Petri net shown in Fig. 1 Conformance deals with comparing an a priori process model with the observed behavior as recorded in the log and aims at detecting inconsistencies/deviations between a process model and its corresponding execution log.

B. Concept Drift:

Concept drift [12] in machine learning and data mining refers to situations when the relation between the input data and the target variable, which the model is trying to predict, changes over time in unforeseen ways. Therefore, the accuracy of the predictions may degrade over time. To prevent that, predictive models need to be able to adapt online, i.e., to update themselves regularly with new data. The setting is typically looped over an infinite data stream as follows: 1) receive new data; 2) make a prediction; 3) receive feedback (the true target value); and 4) update the predictive model. While operating under such circumstances, predictive models are required: 1) to react to concept drift (and adapt if needed) as soon as possible; 2) to distinguish drifts from once-off noise and adapt to changes, but be robust to noise; and 3) to operate in less than data arrival time and use limited memory for storage. In this setting, many adaptive algorithms have been developed.

Concept drift is a relatively young research topic that has gained popularity in data mining and machine learning communities in the last 10 years. Concept drift research primarily has been focusing on two directions:

1) how to detect drifts (changes) online how to keep predictive models up to date. Concept drift has been shown to be important in many applications. The basis for drift detection could be a raw data stream, a stream of prediction errors, and, more rarely, a stream of predictions or a stream of updated model parameters. Two types of concept drift detection approaches have been used: monitoring evolution of a stream or comparing data distributions in two time windows. The cumulative sum (CUSUM) approach is a representative sequential analysis technique for change detection, different extensions to which have been proposed.

III. PROPOSED APPROACH FRAMEWORK AND DESIGN

3.1. Architecture

Over the last two decades many researchers have been working on process flexibility. Ploesser et al. [32] have classified business process changes into three broad categories: 1) sudden; 2) anticipatory; and 3) evolutionary. This classification is used in this paper, but now in the context of event logs.

This approach uses process mining to provide an aggregated overview of all changes that have happened so far. This approach, however, assumes that change logs are available, i.e., modifications of the workflow model are recorded. At this point of time, very few information systems provide such change logs.

Concept drift is in various branches of the data mining and machine learning community. Concept drift has been two types supervised and unsupervised settings and has been shown to be important in many applications. Unlike in data mining and machine learning, where concept drift focuses on changes in simple structures such as variables, concept drift in process mining deals with changes to complex artefact's such as process models describing concurrency, choices, loops, and cancelation.

They work differs from in several ways: 1) this approach constructs an abstract representation of a process unlike ours where we consider features characterizing the traces and 2) this technique is applicable only for change detection whereas our framework is applicable for both change (point) detection and change localization.

3.2 Process Flow:
The various aspects of process change. Initially, we describe change perspectives (control flow, data, and resource). Then, the different types of drift (sudden, gradual, recurring, periodic, and incremental).

A. Perspectives of Change: There are three important perspectives in the context of business processes: 1) control flow; 2) data; and 3) resource. One or more of these perspectives may change over time.

1) Control flow/behavioural perspective: This class of changes deals with the behavioural and structural changes in a process model. Just like the design patterns in software engineering, there exist change patterns capturing the common control-flow changes.
Control flow changes can be classified into operations such as insertion, deletion, substitution, and reordering of process fragments.
2) Data perspective: This class of changes refer to the changes in the production and consumption of data and the effect of data on the routing of cases. For example, it may no longer be required to have a particular document when approving a claim.
3) Resource perspective: This class deals with the changes in resources, their roles, and organizational structure, and their influence on the execution of a process. As example, certain execution paths in a process could be enabled (disabled) upon the availability (non availability) of resources.

B. Nature of Drifts: With the duration for which a change is active, we can classify changes into momentary and permanent. Momentary changes are short lived and affect only a very few cases, whereas permanent changes are persistent and stay for a while,they focus on permanent changes as momentary changes often cannot be discovered because of insufficient data.

1) Sudden drift: This corresponds to a substitution of an existing process M1 with a new process M2, as shown in Fig. 3(a). M1 ceases to exist from the moment of substitution.

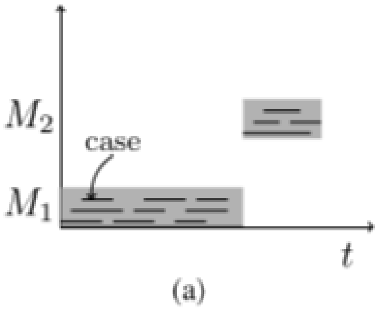


Fig (a). Sudden drift

2) Gradual drift: This refers to the scenario, as shown in Fig. (b) where a current process M1 is replaced with a new process M2. Unlike the sudden drift, here both processes coexist for some time with M1 discontinued gradually.

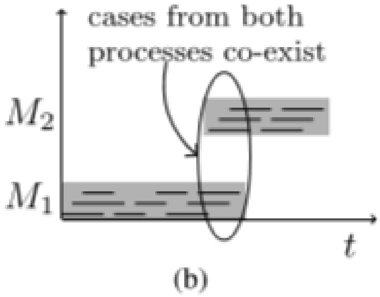


Fig (b): Gradual Drift

3) Recurring drift: This corresponds to the scenario where a set of processes reappear after some time (substituted back and forth), as shown in Fig.(c). It is quite natural to observe such a phenomenon with processes having a seasonal influence.

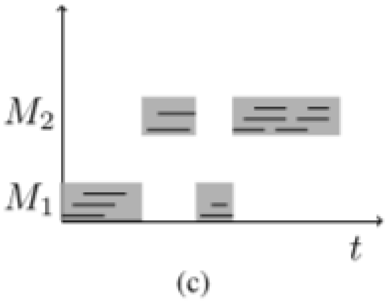


Fig (c): Recurring Drift.

4) Incremental drift: This refers to the scenario where a substitution of process M1 with MN is done via smaller incremental changes, as shown in Fig. (d). This class of

DOI 10.17148/IJARCCE.2016.5969

drifts is more pronounced in organizations adopting an agile BPM methodology and in processes undergoing sequences of quality improvements.

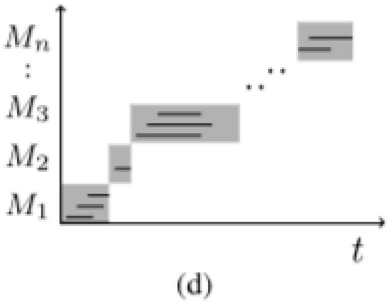


Fig (d): Incremental Drift.

3.3 Mathematical Model:
They present the basic idea for the detection of changes by analysing event logs.
1) A is the set of activities. A+ is the set of all nonempty finite sequences of activities from A.
2) A process instance (i.e., case) is described as a trace over A, i.e., a finite sequence of activities. Examples of traces are abcd and abbbad.
3) Let t = t (1) t (2) t (3) ...t(n) ∈ A+ be a trace over A. |t|=n is the length of the trace t. t(k) is the kth activity in the trace and t (i, j) is the continuous subsequence of t that starts at position i and ends at position j. ti = t (i, |t|) represents the suffix of t that begins at position i.
4) An event log, L, corresponds to a multiset (or bag) of traces from A+. For example, L= [abcd, abcd, abbbad] is a log consisting of three cases. Two cases follow trace abcd and one case follows trace abbbad.
5) N, N0, and R+0 are the set of all natural numbers, the set of all natural numbers including zero, and the set of all positive real numbers including zero, respectively.

IV. FEATURE EXTRACTION

Event logs are characterized by the relationships between activities. Dependencies between activities in an event log can be captured and expressed using the follows (or precedes) relationship, also referred to as causal footprints. For any pair of activities, a, b ∈ A, and a trace t = t (1) t (2) t (3) ...t(n) ∈ A+, we say b follows an if and only if for all 1 ≤ i ≤ n such that t(i) = a there exists a j such that i < j ≤ n and t(j) = b.
In temporal logic notation: (a ⇒ (♦b))
They distinguish between two classes of features:
1) global and
2) local features.

Global features are defined over an event log, whereas local features can be defined at a trace level. With the follows (precedes) relation, we propose two global features: 1) relation type count (RC) and 2) relation entropy (RE), and two local features: 1) window count (WC) and 2) J measure. These features are defined as follows.

(A) RC: The RC with respect to the follows (precedes) relation is a function, f L/RC: A→ N0×N0×N0, defined over the set of activities A. f L/RC of an activity, x ∈ A, with respect to the follows (precedes) relation over an event log L is the triple {cA, cS, cN} where {cA, cS, and cN} are the number of activities in A.
(B) RE: The RE with respect to the follows (precedes) relation is a function, f L/RE: A→ R+ 0, defined over the set of activities. F L/RE of an activity, x ∈ A with respect to the follows (precedes) relation is the entropy of the RC metric.
(c)) WC: Given a window of size l ∈ N, the WC with respect to follows (precedes) relation is a function,
F lM/WC: A×A→ N0, defined over the set of activity pairs.
(D) J measure: They have proposed a metric called J measure based on to quantify the information content (goodness) of a rule. They adopt this metric as a feature to characterize the significance of relationship between activities. The basis lies in the fact that we can consider the relation b follows an as a rule: if activity an occurs, then activity b will probably occur. The J measure with respect to follows (precedes) relation is a function
f l, t/J: A×A→ R+ defined over the set of activity pairs and a given window of length l ∈ N.

V. FRAMEWORK

Event logs are characterized by the relationships between activities. Dependencies between activities in an event log can be captured and expressed using the follows (or precedes) relationship, also referred to as causal footprints. For any pair of activities, a, b ∈ A, and a trace t = t (1) t (2) t (3) ...t(n) ∈ A+, we say b follows an if and only if for all 1 ≤ i ≤ n such that
t(i) = a there exists a j such that i < j ≤ n and t(j) = b. In temporal logic notation: (a ⇒ (♦b)).

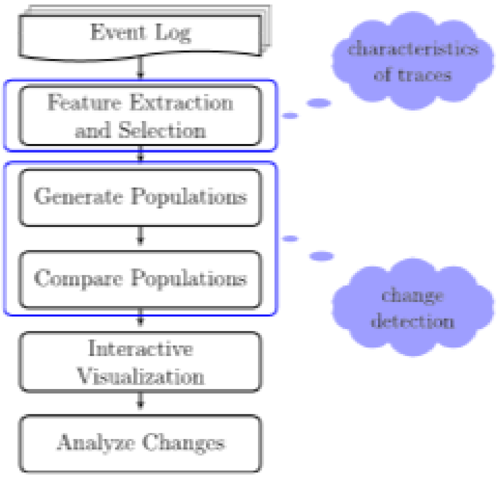

Event Log
Feature Extraction and Selection
Generate Populations
Compare Populations
Interactive Visualization
Analyze Changes
characteristics of traces
change detection

Fig. Framework for handling

They distinguish between two classes of features:
1) Global and
2) Local features.

DOI 10.17148/IJARCCE.2016.5969

Global features are defined over an event log, whereas local features can be defined at a trace level. With the follows (precedes) relation, they propose two global features: 1) relation type count (RC) and 2) relation entropy (RE), and two local features: 1) window count (WC) and 2) J measure.

1) RC:
The RC with respect to the follows (precedes) relation is a function,
f: A→ N0×N0×N0

f of an activity, x ∈A, with respect to the follows (precedes) relation over an event log L is the triple {cA, cS, cN} where {cA, cS, and cN} are the number of activities in A that always, sometimes, and never follows (precedes) x, respectively, in the event log L.
2) RE:
The RE with respect to the follows (precedes) relation is a function,
f: A→ R+ 0,
defined over the set of activities f of an activity, x ∈ A with respect to the follows (precedes) relation is the entropy of the RC metric.
3) WC:
Given a window of size l ∈ N, the WC with respect to follows (precedes) relation is a function,
f: A×A→ N0,
defined over the set of activity pairs. Given a trace t and a window of size l, letSl, t(a) be the bag of all subsequence’st (i, i +l−1), such that t(i)=a.3
4) J Measure:
J measure with respect to follows (precedes) relation is a function
f: A×A→ R+
defined over the set of activity pairs and a given window of length l ∈ N. Let pt.(a) and pt.(b) are the probabilities of occurrence of activities an and b, respectively, in a trace t.

VI. WORK DONE

In this section we are discussing the practical environment, scenarios, performance metrics used etc.

6.1. Input:
In this Training and Testing Image is the input for our practical experiment.

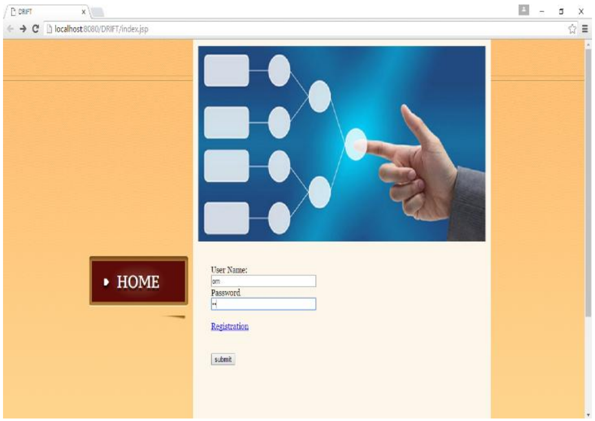
6.2. Hardware Requirements:

- Processor : -P-IV– 500 MHz to 3.0 GHz
- RAM : - 1GB
- Disk : -20 GB
- Monitor : -Any Color Display
- Standard Keyboard and Mouse

6.3. Software Requirements:

- Operating System : -Windows 7/XP
- Programming Languages : - Java
- Database Server : - My Sql
- Web server : - Apache Tomcat

6.4 Results of Practical Work:
Following figures are showing results for practical work which is done. Following figure showing the main screen. That takes the input data set,



User Name:
om
Password
**
Registration

Fig.1 Take the input data as User Name and Password.

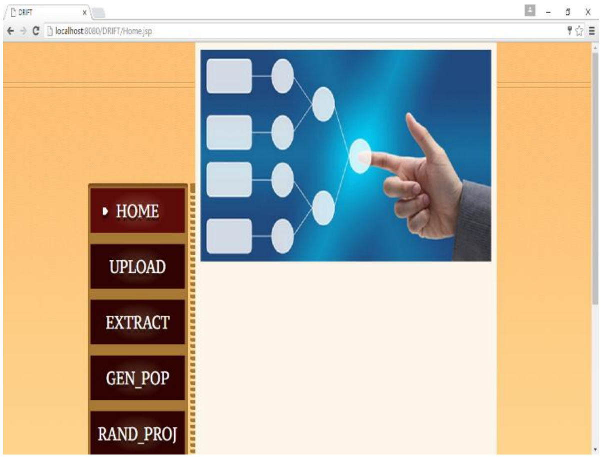
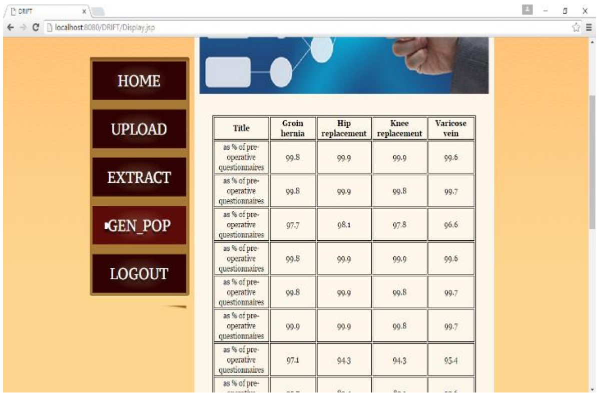


Fig. 2 Home Screen



Title	Groin hernia	Hip replacement	Knee replacement	Varicose vein
as % of pre-operative questionnaires	99.8	99.9	99.9	99.6
as % of pre-operative questionnaires	99.8	99.9	99.8	99.7
as % of pre-operative questionnaires	97.7	98.1	97.8	96.6
as % of pre-operative questionnaires	99.8	99.9	99.9	99.6
as % of pre-operative questionnaires	99.8	99.9	99.8	99.7
as % of pre-operative questionnaires	99.9	99.9	99.8	99.7
as % of pre-operative questionnaires	97.1	94.3	94.3	95.4
as % of pre-	[illegible]	[illegible]	[illegible]	[illegible]

Fig. 3 Gen_POP Screen

VII. CONCLUSION

We have studied the topic of concept drift in process mining, i.e., analysing process changes based on event

DOI 10.17148/IJARCCE.2016.5969

logs. They proposed feature sets and techniques to effectively detect the changes in event logs and identify the regions of change in a process. Their initial results show that heterogeneity of cases arising because of process changes can be effectively dealt with by detecting concept drifts. Once change points are identified, the event log can be partitioned and analysed. This is the first step in the direction of dealing with changes in any process monitoring and analysis efforts. They have considered changes only with respect to the control flow perspective manifested as sudden and gradual drifts.

REFERENCES

[1] R. P. Jagadeesh Chandra Bose, Wil M. P. van der Aalst, Indre˙ Žliobait˙e, and Mykola Pechenizkiy

[2] “Dealing with Concept Drifts in Process Mining”

[3] IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 25, NO. 1, JANUARY 2014.

[4] All-in-one Permit for Physical Aspects: (Omgevingsvergunning) in a Nutshell [Online].Available:http://www.answersforbusiness.nl/ regulation/all-in-one-permit-physical-aspects.

[5] United States Code. (2002, Jul.). Sarbanes-Oxley Act of 2002, PL 107-204, 116 Stat 745 [Online]. Available:http://files.findlaw.com/ news.findlaw.com/cnn/docs/gwbush/sarbanesoxley072302.pdf

[6] W. M. P. van der Aalst, M. Rosemann, and M. Dumas, “Deadline-based escalation in process-aware information systems,” Decision Support Syst., vol. 43, no. 2, pp. 492–511, 2011.

[7] N. Mulyar, “Patterns for process-aware information systems: An approach based on coloured Petri nets,” Ph.D. dissertation, Dept. Comput. Sci., Univ. Technol., Eindhoven, The Netherlands, 2009.

[8] B. Weber, S. Rinderle, and M. Reichert, “Change patterns and change support features in process-aware information systems,” in Proc. 19th Int., 2007, pp. 574–588.

[9] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, and W. M. P. van der Aalst, “Process flexibility: A survey of contemporary approaches,” in Proc. Adv. Enterprise Eng. I, 2008, pp. 16–30.

[10] L. L. Minku, A. P. White, and X. Yao, “The impact of diversity on online ensemble learning in the presence of concept drift,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 5, pp. 730–742, May 2010.

[11] D. Brzezinski and J. Stefanowski, “Reacting to different types of concept drift: The accuracy updated ensemble algorithm,” IEEE Trans. Neural Netw. Learn. Syst., Apr. 2013, doi: 10.1109/TNNLS.2013.2251352.

[12] A. Bifet and R. Kirkby. (2011). Data Stream Mining: A Practical Approach, University of Waikato, Waikato, New Zealand [Online]. Available:http://www.cs.waikato.ac.nz/~abifet/MOA/StreamMining .pdf

[13] I. Žliobaite˙, “Learning under concept drift: An Overview,” CoRR, vol. abs/1010.4784, 2010 [Online]. Available: http://arxiv.org/abs/1010.4784

[14] J. Schlimmer and R. Granger, “Beyond incremental processing: Tracking concept drift,” in Proc. 15th Nat. Conf. Artif. Intell., vol. 1. 1986, pp. 502–507.

BIOGRAPHY

Trupti Kakkad receives her Bachelor in engineering degree in Information Technology from Rajiv Gandhi College of Engineering and Research Technology, Chandrapur in 2007 - 2008. She has a work experience of 5 years in TATA Consultancy Services.

DOI 10.17148/IJARCCE.2016.5969