

# Semi-supervised Classification in Data Mining for Concept Impulsion Detection and Management

G Swamy<sup>[1]</sup>, Dr. Om Prakash<sup>[2]</sup>

<sup>[1]</sup>PhD Scholar, Department of Computer Science and Engineering, OPJS University, Rajasthan, INDIA

<sup>[2]</sup>Professor, Department of Computer Science and Engineering, OPJS University, Rajasthan, INDIA

**Abstract:** Concept impulsion is a vital problem for any information investigation situation including transiently requested information. In predictive analytics and machine learning, the concept impulsion means that the measurable properties of the objective variable, which the model is endeavoring to anticipate, change after some time in unanticipated ways. This causes issues in light of the fact that the expectations turn out to be less precise over the long time. The term concept refers to the amount to be indicated. In the most recent decade Process mining emerged as a method that uses the logs of data arrangement with a specific end goal to mine, break down and upgrade the procedure measurement. Concept impulsion in the process is localized by applying statistical hypothesis testing methods. The proposed technique is tested and approved on few of the real-life and artificial process logs, results about acquired are promising toward effectively confining the sudden approach impulsion in process-log, results obtained are promising in the direction of efficiently localizing the sudden concept impulsion in process-log. We present the first online mechanism for detecting and managing concept impulsion, which is based on abstract interpretation and sequential sampling, together with recent learning techniques on data streams. We propose a Semi-supervised classification algorithm for data streams with Revenant concept Impulsion and Limited Tagged data, called RILT, in which, a decision tree is adopted as the classification model. The discovered process models can be used for a variety of analysis purposes. Finally, we investigated the challenges, different process mining algorithms, classification of process mining techniques.

## I. INTRODUCTION

BUSINESS processes are nothing more than logically related tasks that use the resources of an organization to achieve a define 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 enactments, for example, the WABO act [1] and the Sarbanes– Oxley Act [2], intense varieties in amount and order, occasional impacts, natural adversities and emergency [3], etc, are likewise compelling associations to change their procedures. For

instance, administrative and insurance associations diminish the part of cases being checked when there is excessively of work in the pipeline. As 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. In this manner, adaptability and changes has been examined top to bottom with regards to business process administration (BPA). For instance, process-aware data systems (PADSs) [4] have been stretched out to have the capacity to adaptability adjust to changes all the while. Best in class work flow administration (WFM) and BPA frameworks [5] give such adaptability. In addition, in forms not driven by WFM/BPM frameworks, (for example, the utilization of medical systems) there is much more flexibility as procedures is controlled by individuals instead of data systems. A significant number of the present data systems are recording a plenty of occasion logs. Process mining is a moderately new research method went for finding, observing, and enhancing genuine procedures by removing learning from occasion logs [6]. Despite the fact that adaptability and change have been considered top to bottom with regards to WFM and BPA systems, contemporary process mining methods accept the procedures to be in a consistent state. For instance, while finding a procedure show from occasion logs, it is expected that the procedure toward the start of the recorded period is the same as the procedure toward the finish of the recorded period. Using ProM, we have analyzed processes in more than 100 organizations. These practical experiences show that it is very unrealistic to assume that the process being studied is in a steady state. As mentioned earlier, processes may change to adapt to changing circumstances. Theory cluster refers to the circumstance in which the procedure is changing while at the same time being investigated. There is a requirement for strategies that arrangement with such second-arrange flow. Breaking down such changes is of most extreme significance when supporting or enhancing operational procedures and to acquire an exact understanding on process executions at any moment of time. When managing theory clusters in process mining, the accompanying three fundamental difficulties emerge. In this paper, we concentrate on two of the difficulties: 1) change (point) recognition and change restriction and 2) characterization. We define different features and propose a framework for dealing with these two problems from a control-

flow perspective. Finally, we demonstrate the guarantee of the systems proposed in this paper on a real-life contextual investigation from a substantial government e-market in India.

## II. RELATED WORK

Being a relatively young research discipline, several process mining challenges remain to be addressed. The process mining manifesto [7] lists 11 challenges. The fourth challenge is dealing with concept drift and, thus far, a little work has been done on this highly relevant topic [8-9]. Concept drift [10] in machine learning and data mining refers to situations at the point when the connection between the information and the objective variable, which the model is endeavoring to anticipate, changes after some time in unexpected ways. Therefore, the accuracy of the predictions may degrade over time. To keep that, prescient models should have the capacity to adjust on the web, i.e., to refresh themselves frequently with new information. The setting is ordinarily circled over an infinite information stream as takes after: 1) get new information; 2) make an expectation; 3) get criticism (the genuine target esteem); and 4) refresh the prescient model. While operating under such circumstances, predictive models are required: 1) to respond to theory cluster (and adjust if necessary) at the earliest opportunity; 2) to recognize clusters from once-off interference and adjust to changes, however be strong to disturbance; and 3) to work in under information arrival time and utilize constrained memory for capacity. In this setting, numerous versatile calculations have been created (e.g., overviews [11], [12]). Throughout the latest two decades various examiners have been wearing down process versatility. In [13] and [14] accumulations of normal change designs are depicted. In [15] and [16] expansive logical orders of the distinctive versatility philosophies and segments are given. Ploesser et al. [17] have portrayed business process changes into three general orders: 1) sudden; 2) eager; and 3) change. This characterization is utilized as a part of this paper, however now with regards to occasion logs. Despite the many publications on flexibility, most process mining techniques assume a process to be in a steady state. A notable exception is the approach in [18].

This approach uses process mining to provide an aggregated overview of all changes that have happened so far. This approach, in any case, assumes that change logs are accessible, i.e., adjustments of the work flow display are recorded. Now at this time, very less data systems give such change logs. Subsequently, this paper concentrates on theory cluster in process mining accepting just an occasion log as information. The theme of theory cluster is very much concentrated in different branches of the information mining and machine learning group. Theory cluster has been examined in both managed and unsupervised settings and has been appeared to be essential in numerous applications [10], [12], [19]–[22]. The problem of concept drift, however, has not been considered in the process mining setting. Not at all like in information mining

and machine realizing, where theory cluster concentrates on changes in basic structures, for example, factors, theory cluster in process mining manages changes to complex artifacts, for example, process models depicting simultaneousness, decisions, circles, and cancelation. In spite of the fact that encounters from information mining and machine learning can be utilized to research theory cluster in process mining, the complicity of process models and the idea of process change pose new difficulties. This paper expands the work introduced in [10]. In this extended paper, we present the point of theory cluster in process mining and present the essential thought and the highlights catching the qualities of follows in an occasion sign in a more thorough way. In addition, this extended paper gives a universal structure to taking care of theory clusters in process mining and shows subtle elements on the acknowledgment of the approach in the ProM system. Furthermore, this paper reports new experimental results of the proposed approach. More specifically, in this extended paper, we study the influence of population size on change point detection and the applicability of the approach in dealing with gradual drifts. In addition, we present the results of applying the approach on a real-life case study from a large Dutch municipality. Recently, Carmona and Gavalda [11] have proposed an online technique for detecting process changes. They first created an abstract representation of the process in the form of polyhedra using the prefixes of some initial traces in the event log. Resulting elements are examined and evaluated whether they exist in the polyhedra or not. If a sample lies within the polyhedra, it is considered to be from the same process. If significant number of samples lies outside the polyhedra, a process change is said to be detected. This work differs from our approach 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 system is appropriate just for change location though our structure is pertinent for both change (point) recognition and change restriction. Furthermore, the tool support provided by the authors has some limitations in its applicability. The tool does not detect change points and does not work on logs with multiple process changes, i.e., it does not detect the presence/absence of multiple changes and does not report when (the trace index) process changes have happened. The tool just reports that a change exists and terminates (if changes exist) and does not terminate if no changes exist. In contrast, our tool can handle multiple process changes and can detect both the presence of and the points of change in addition to being able to assist in change localization.

## III. PROPOSED SYSTEM

Proposed algorithm to be presented in this section aims to handle relevant concept impulsing data streams with unlabeled data the processing flow of RILT with the incoming of streaming data untagged data are tagged at leaves using a clustering strategy and the information of untagged data is

reused for the growing of the tree the revenant concept impulsing detection is installed using concept clusters maintained at leaves to avoid the space overflow over-fitting with the continuously growing of the tree, a pruning mechanism is adopted when reaching a period Lastly to track the performance of the current classification model, prediction results are evaluated periodically in the Prequential estimation Technique details involved in this processing.

### RILT Algorithm

**Input:** A Stream of instances= $I$  ; Minimum number of split examples =  $n_{min}$  ; period of detection =  $PD$  ; period of pruning =  $PP$  ; output period increment =  $OP$

**Output:** Classification Error

### Procedure RILT $\{I, n_{min}, PD, PP, OP\}$

Step 1: Create a tree 'T' with number of leafs 'l';  
 Step 2: for each instance- $e \in E$ ;  
 Step 3: store the corresponding information by sorting 'e' into an available leaf 'l';  
 Step 4: if the arrived instances count at 'l' meets  $n_{min}$   
 Step 5: Tag the untagged instances at leaf 'l' in k-Means;  
 Step 6: Install a split test and grow children leaves;  
 Step 7: if arrived number of instances %  $PD = 0$   
 Step 8: Detect revenant concept impulsing using history concept clusters ad new ones;  
 Step 9: If number of instances arrived %  $PP = 0$   
 Step 10: Install the bottom-up search and pune subtrees regarding the classification error;  
 Step 11: If number of instances arrived %  $OP = 0$   
 Step 12: Report the classification result using Prequential estimation;

To exploit untagged data, we adopt k-Means to create concept clusters and implement tagging, because k-Means is a simple and efficient clustering algorithm for numerical attributes. The clustering algorithm will be activated if there are new labeled information at the present leaf Based on these created concept groups the greater part class technique is utilized to label untagged information.

#### A. Clustering Algorithms based Concept impulsing/Drift

Use clustering for detecting concept drifts every trace is converted to a vector of maximal repeats. As characterized a maximal repetition in a sequence  $s$  is characterized as a subsequence that happens in a maximal combine. A maximal match in an arrangement  $s$  is a subsequence shows in  $s$  at two particular positions and with the end goal that the component to the quick left (ideal) of the appearance of position is not the same as the component to one side (ideal) of the indication. The time dimension is further added to each vector. Agglomerative Hierarchical Clustering (AHC) with the base fluctuation standard is utilized as the bunching calculation. There is no

openly accessible usage of the calculation to test or incorporate. Three fairly basic manufactured cases were utilized to test the calculation the deliberate metric was the exactness of clustering. Their algorithm had assigned the traces to correct clusters in 70% to 100% cases.

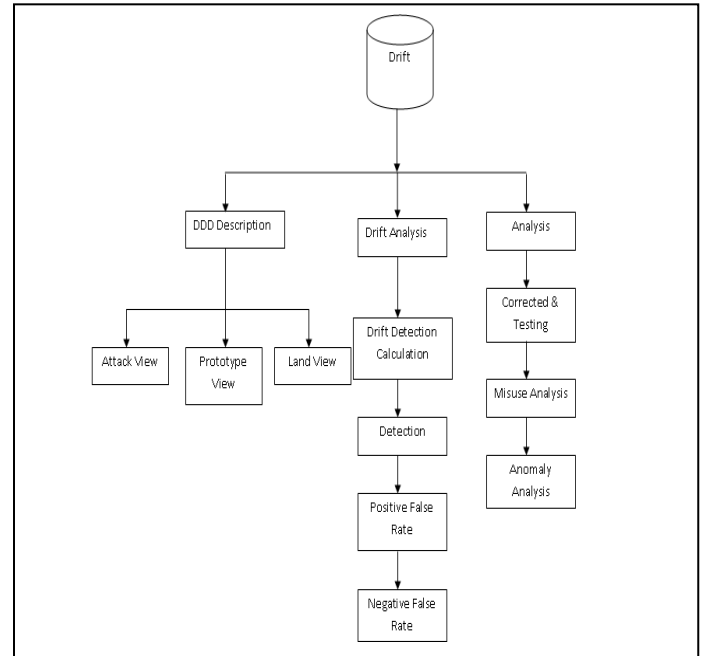


Fig.1: Proposed System Architecture

#### Algorithm 1 Change Point Detection

1: Let  $P_1$  and  $P_2$  be the two populations where we have detected a change (i.e., its hypothesis test's  $p$ -value  $< \hat{p}$ ).  
 2: Split the two populations  $P_1$  and  $P_2$  into halves,  $P_{11}$  and  $P_{12}$  for  $P_1$  and  $P_{21}$  and  $P_{22}$  for  $P_2$ .  
 3: Apply hypothesis tests on the left ( $P_{11}$  and  $P_{12}$ ), center ( $P_{12}$  and  $P_{21}$ ), and right ( $P_{21}$  and  $P_{22}$ ) population pairs illustrated. Let  $p_{left}$ ,  $p_{center}$ , and  $p_{right}$  be their respective  $p$ -values.  
 4: Let  $p_{min} = \min\{p_{left}, p_{center}, p_{right}\}$ . Let  $P_{min}^1$  and  $P_{min}^2$  be the corresponding populations of  $p_{min}$ .  
 5: If  $p_{min} < \hat{p}$ , set  $P_1 = P_{min}^1$  and  $P_2 = P_{min}^2$ , goto step 1, else return the index/time point corresponding to the trace at end of  $P_{min}^1$  as the change point.

#### B. Adaptive Windows

The calculation is exceedingly subject to the picked population capacity. If this parameter is too little then the plot can contain a great deal of disturbance therefore either the noise can be misconstrued as theory clusters will be left undetected between arbitrary troughs where theory clusters. If the population estimate turns out to be too huge then the time complicity of the calculations declines and a few clusters may end up plainly undetected.

### Simulation Verification

The experiments demonstrate that the features and the framework proposed in this paper for handling concept impulsions show significant promise in detecting behavioral changes by analyzing event logs. To simulate the concept impulsion process in data mining, we created the data log consisting of various electrical appliances from the government e-market place in India for the duration of 5 years i.e., from 2013 to 2017 years, which equals 60 logs. Total number of products considered here is 8. We had shown that the mining classification with concept impulsion with the red zone, which indicates that the sudden change occurs in product selling i.e., gradual increment or decrements to assess the product monthly and yearly review.

#### Algorithm 2 Change Detection using Adaptive Windows

**Require:** a minimum population size  $w_{min}$ , a maximum population size  $w_{max}$ ,  $p$ -value threshold  $\hat{p}$ , a step size  $k$ , and a data stream of values  $D$

- 1: let  $P_{left}$  and  $P_{right}$  be two populations of size  $w_{min}$  with  $P_{right}$  starting at the first index after the end of  $P_{left}$
- 2: **repeat**
- 3: Apply hypothesis test over  $P_{left}$  and  $P_{right}$ . Let  $p$  be its  $p$ -value
- 4: **if**  $p < \hat{p}$  **then**
- 5: identify the change point within  $P_{left}$  and  $P_{right}$  using Algorithm 1
- 6: create two new populations  $P'_{left}$  and  $P'_{right}$  of size  $w_{min}$  with  $P'_{left}$  starting at the first index after the end of  $P_{right}$  and  $P'_{right}$  starting at the first index after the end of  $P'_{left}$ . Set  $P_{left} = P'_{left}$  and  $P_{right} = P'_{right}$ .
- 7: **else**
- 8: Extend the left and right populations by step size  $k$ . Reassign the right population to start at the first index after the end of the extend left population  $P_{left}$ .
- 9: **if** the size of the population is  $\geq w_{max}$  **then** discard the left population  $P_{left}$ . Split the right population  $P_{right}$  into two halves and use them as the left and right populations.
- 10: **end if**
- 11: **end if**
- 12: **until** the end of  $P_{right}$  doesn't reach the end of  $D$

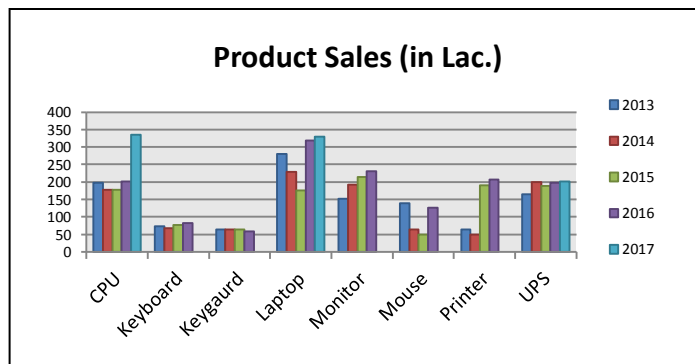


Fig.2: Product mining analysis

### IV. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented the point of theory impulsion in process mining, i.e., investigating process changes in view of occasion logs. We proposed feature sets and techniques to effectively detect the changes in event logs and identify the regions of change in a process. Our initial results show that heterogeneity of cases arising because of process changes can be effectively dealt with by detecting impulsions. Once change points are identified, the event log can be partitioned and analyzed. This is the first step in the direction of dealing with changes in any process monitoring and analysis efforts. We have considered changes only with respect to the control flow perspective manifested as sudden and gradual impulsions.

1) **Change-pattern specific features:** In this, we presented very generic features (based on follows/precedes relation). These features are neither complete nor sufficient to detect all classes of changes. An important direction of research is to define features catering to different classes of changes and to investigate their effectiveness. A taxonomy/classification of change patterns and the appropriate features for detecting changes with respect to those patterns are needed.

2) **Holistic approaches:** In this, we considered about thoughts on change recognition and limitation with regards to sudden and continuous changes to the control-flow point of view of a procedure. The data and resource perspectives are also, however, equally important. Features and techniques that can enable the detection of changes in these other perspectives need to be discovered. Furthermore, there could be instances where more than one perspective (e.g., both control and resource) change simultaneously. Hybrid approaches considering all aspects of change holistically need to be developed.

3) **Recurring impulsions:** When dealing with recurring impulsions, in addition to change point detection and change localization, it is important to identify the variant(s) that recur. This requires robust metrics to assess the similarity between process variants and/or event logs.

### V. REFERENCES

- [1] (2010). All-in-one Permit for Physical Aspects: (Omgevingsvergunning) in a Nutshell
- [2] United States Code. (2002, Jul.). Sarbanes-Oxley Act of 2002, PL 107-204, 116 Stat 745
- [3] 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.
- [4] M. Dumas, W. M. P. van der Aalst, and A. H. M. Ter Hofstede, ProcessAware Information Systems: Bridging People and Software Through Process Technology. New York, NY, USA: Wiley, 2005.

- [5] W. M. P. van der Aalst and K. M. van Hee, *Workflow Management: Models, Methods, and Systems*. Cambridge, MA, USA: MIT Press, 2004.
- [6] W.M.P.vanderAalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. New York, NY, USA: Springer-Verlag, 2011.
- [7] F. Daniel, S. Dustdar, and K. Barkaoui, "Process mining manifesto," in *BPM 2011 Workshops*, vol. 99. New York, NY, USA: Springer-Verlag, 2011, pp. 169–194.
- [8] R. P. J. C. Bose, W. M. P. van der Aalst, I. Žliobaitė, and M. Pechenizkiy, "Handling concept drift in process mining," in *Proc. Int. CAiSE*, 2011, pp. 391–405.
- [9] J. Carmona and R. Gavalda, "Online techniques for dealing with concept drift in process mining," in *Proc. Int. Conf. IDA*, 2012, pp. 90–102.
- [10] J. Schlimmer and R. Granger, "Beyond incremental processing: Tracking concept drift," in *Proc. 15th Nat. Conf. Artif. Intell.*, vol. 1. 1986, pp. 502–507.
- [11] 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>
- [12] I. Žliobaitė, "Learning under concept drift: An Overview," *CoRR*, vol. abs/1010.4784, 2010 [Online]. Available: <http://arxiv.org/abs/1010.4784>
- [13] N. Mulyar, "Patterns for process-aware information systems: An approach based on colored Petri nets," Ph.D. dissertation, Dept. Comput. Sci., Univ. Technol., Eindhoven, The Netherlands, 2009.
- [14] B. Weber, S. Rinderle, and M. Reichert, "Change patterns and change support features in process-aware information systems," in *Proc. 19<sup>th</sup> Int.*, 2007, pp. 574–588.
- [15] G. Regev, P. Soffer, and R. Schmidt, "Taxonomy of flexibility in business processes," in *Proc. 7th Workshop BPMDS*, 2006, pp. 1–4.
- [16] 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.
- [17] K. Ploesser, J. C. Recker, and M. Rosemann, "Towards a classification and lifecycle of business process change," in *Proc. BPMDS*, vol. 8. 2008, pp. 1–9.
- [18] C. W. Günther, S. Rinderle-Ma, M. Reichert, and W. M. P. van der Aalst, "Using process mining to learn from process changes in evolutionary systems," *Int. J. Business Process Integr. Manag.*, vol. 3, no. 1, pp. 61–78, 2008.
- [19] M. van Leeuwen and A. Siebes, "StreamKrimp: Detecting change in data streams," in *Proc. Mach. Learn. Knowl. Discovery Databases*, 2008, pp. 672–687.
- [20] I. Žliobaitė and M. Pechenizkiy. (2010). *Handling Concept Drift in Information Systems* [Online]. Available: [http://sites.google.com/site/zliobaite/CD\\_applications\\_2010.pdf](http://sites.google.com/site/zliobaite/CD_applications_2010.pdf)
- [21] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. 2003, pp. 226–235.
- [22] 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.