Master Thesis

Interactive events identification for process data based on change point detection

Submitted by:	Prashant Pratik
Matriculation no.:	1271359
First examiner:	Prof. Dr. Andreas Pech
Second examiner:	Prof. Dr. Heiko Hinkelmann
Company supervisor:	Dr. Ruomu Tan
Date of start:	01.10.2021
Date of submission:	01.03.2022

Statement

I confirm that I have written this thesis on my own. No other sources were used except those referenced. Content which is taken literally or analogously from published or unpublished sources is identified as such. The drawings or figures of this work have been created by myself or are provided with an appropriate reference. This work has not been submitted in the same or similar form or to any other examination board.

_____

Date, Signature of the Student

Content

1 Introduction

With the advancement in sensor technologies, huge amounts of industrial data are being generated every day, and most of the data are present in the form of time series data. These time series data generated in the process industry are also known as process data. There is a need to get some useful information out of these data. One of the possible use cases is the identification of the different events associated with these time series process data. In industrial processes, the operating conditions can change from time to time. So, it becomes important to keep track of these changing conditions or events [1]. This thesis presents approaches to achieve the identification of these events associated with the process data interactively by incorporating the feedback from process experts using the change point detection algorithm.

Based on the way of production, an industrial process can either be classified as a continuous process or a batch process. A continuous process consists of a constant flow of input materials and the produced product is regularly removed for creating space for a new product. On the other hand, a batch process produces materials in batches where a batch can be defined as a single run of a sequence of operations that should be executed in a specific order to produce certain materials [2]. In this thesis, the focus is batch processes because the identification of the operations and events in batch processes is essential for labeling and analysis of the time series data.

For the analysis of batch process data, many algorithms and techniques are available, and change point detection is one of those algorithms. Change point detection can be described in general terms as the identification of times when the probability distribution of a stochastic process or time series segment changes. But for process data, a change point can be defined as a point in time where an event or a phase in the process starts or ends. And based on these detected change points through the change point detection algorithm, an event or a phase in a process data can be identified. Moreover, change point detection can be performed either in real-time or after collecting all the data. When the detection is performed in real-time, it is known as online change point detection and when it is performed after collecting the data, it is known as offline change point detection [1] [3]. All the work in this thesis is based on offline change point detection.

This thesis addresses the problem of identifying events associated with the process data along with providing different possible solutions for it using the change point detection algorithm. In this thesis, first, the unsupervised approach for the change point detection is experimented with and its drawbacks are discussed, then, the interactive change point detection approach is proposed, experimented with and the results are discussed, and finally, a prototype for the interactive process is also implemented.

1.1 Aim and Motivation

Identification of batch events is important for the analysis of batch process data. For some batch processes, the batch events are not recorded. Moreover, inaccurate information about the events associated with the batch process data can be found. Although there are many machine learning (ML) algorithms available that can be used to identify the events associated with time series data, most of them are based on supervised learning and hence, require labeled data and it is a very strenuous task to label these huge amounts of data manually. However, there are other ways in which this task can be achieved such as an unsupervised ML algorithm along with user feedback where user and algorithm can work together to label the huge sets of data interactively or more specifically, create artificial labels for the batch events. In the case of process data, user feedback can play an important role as users are process experts and their feedback can improve the performance of the unsupervised algorithm by helping them to identify the events more accurately.

Therefore, the aim of this Master Thesis is to study different approaches and implement a prototype that can provide suitable visualizations of the given time series process data to the user which can then interactively identify the events associated with these process data based on change point detection algorithm with the help of user feedback.

1.2 Challenges

Identification of the events in a batch process is not an easy task due to its dynamic nature and different complexities involved in it. There are multiple batches in a batch process and within each batch, there is a sequence of operations that can occur parallelly and overlap with one another. Moreover, these batch operations are very flexible and can change process conditions and production volumes. Therefore, batch processes, in general, are non-linear and non-stationary in nature which makes it more difficult for a machine learning algorithm to predict its behavior [4].

Apart from the complex characteristics of the batch process, measurements of process variables from different sensors are also prone to disturbances and noise. Furthermore, in a real production dataset, there could be the presence of missing timestamps in the recorded time series data. These real-life problems make the batch process data analysis even more challenging.

Throughout this thesis, most of the above-mentioned challenges are addressed and different solutions are proposed, experimented with, evaluated, and discussed.

1.3 Work Plan and Requirements

The target of this thesis is to research, experiment with different ways to identify events in a process data interactively using user feedback based on change point detection, implement a prototype for this interactive process, summarize the results and conclude. To achieve this target, a certain work plan is required. In this section, involved tasks, requirements, associated research questions, and the structure of the thesis are described in detail.

1.3.1 Tasks involved

The detailed requirements for this thesis and the involved tasks are stated below:

- Literature review.
- Understanding the Change Point Detection algorithm along with its different parametric settings.
- Data understanding for various real-life and simulated datasets containing time series process data.
- Experimenting and testing the unsupervised CPD algorithm on a simulated dataset.
- Evaluating and comparing the results for unsupervised CPD performance using evaluation metrics.
- Experimenting and testing the interactive CPD algorithm on various datasets.
- Providing suitable visualizations of time series data, change points, and detected events to the user.
- Incorporating the initial user feedback to interactive CPD algorithm for events identification.
- Evaluating and comparing the results for interactive CPD performance using visualizations and evaluation metrics.
- Incorporating the user feedback to interactive CPD algorithm for correction/update of results.
- Developing a UI prototype for this complete interactive process.
- Testing the prototype on various datasets.

1.3.2 Associated Research Questions

Throughout this thesis, the following research questions are answered:

- How to achieve the identification of the events for process data through interactive CPD?
- How to provide the best visualization and incorporate user feedback into interactive CPD to obtain both initial and updated results?
- How to evaluate and measure the results of interactive CPD?

The main research work of this thesis is defined by the first question and creates room for experiments with different approaches. The second question suggests the need for a prototype with a user interface along with tuning of different parameters. Finally, the performance of different approaches is discussed through the third question.

1.3.3 Thesis Structure

This thesis work is structured in six main chapters: chapter 1 gives the introduction to the topic and challenges of solving the problem of events identification for process data. Chapter 2 reviews the existing solutions in the literature as well as the change point detection algorithm for batch process data analysis. In chapter 3, first, the experiments with the unsupervised change point detection for identifying events on the simulated dataset are performed, their results and drawbacks are discussed, and then, different approaches to solve the identification of events through interactive change point detection along with the implementation of a prototype for the interactive process are discussed in detail. Different experiments on the simulated and the real-life datasets, their results, a summary of findings, and future works are presented and discussed in chapter 4 and chapter 5 respectively. Finally, a conclusion is provided in chapter 6.

2 Background

In this chapter, the necessary background information about the change point detection and currently used algorithms for batch process data analysis are provided. There are three sections in this chapter: the first section describes the batch process. Then, different algorithms are discussed in the second section. Finally, the change point detection algorithm and its application in batch process data analysis are explained in the third section.

2.1 Batch Process

In process industries, materials are either produced through a continuous process or a batch process. When a constant flow of material is maintained between every step of the process without any break in time, then, the materials are produced through a continuous process. But through a batch process, the materials are produced in batches where a batch can be defined as a single run of a sequence of operations that should be executed in a specific order [2].

A batch process can be described by the periodic repetition of pre-defined batch operations. Moreover, an event or a phase is associated with each batch operation. For example, in the chemical industry, these events or phases could be the loading of reactants, discharge, cleaning of vessels, etc. Apart from chemical industries, there are many sectors where batch processes are used for the production of materials such as pharmaceuticals, food industries, semiconductors, etc. Along with an event or a phase, there could be different process times associated with each batch operation ranging from seconds to years depending upon the industrial sector [4].

The main advantage of these batch operations is that they are highly flexible which opens room for better control over the batch production by changing process conditions and production volumes. But from the data analysis point of view, it becomes very challenging because this makes the batch processes dynamic, non-linear, and non-stationary in nature. Therefore, proper data-driven techniques and methods with higher flexibility are required for data analysis of batch processes as compared to continuous processes [4].

2.2 Algorithms for Batch Process Data Analysis

Batch processes are dynamic in nature and due to this reason, different data-driven techniques, methods, and algorithms have been proposed and applied for batch process data analysis. Tasks that are conducted in the scope of batch process data analysis are multivariate statistical analysis, process monitoring, fault detection, quality prediction, phase classification, and many others [4]. In this section, some of the algorithms which are used for these tasks are discussed.

2.2.1 Multivariate Statistical Analysis

Multivariate statistical analysis for batch process data is mostly performed through principal component analysis (PCA) and projection to latent structures/partial least squares (PLA) methods. These methods come under dimensionality reduction techniques. In PCA, a lower dimension space without losing any useful information is found by capturing the maximum amount of variance in an input data matrix. But in PLS, the lower dimension space is found for both input and output data matrices to best predict the output data matrix by capturing the maximum amount of covariance between input and output data matrices [5]. Moreover, an extension to these methods is also used for three-dimensional data. These are multiway principal component analysis (MPCA) and multiway projection to latent structures/partial least squares (MPLA). MPCA can be used to examine the process variability, while to examine the relationship between process trajectories and product quality variables, MPLA can be used [5] [6]. Additionally, there are other variants of PCA and PLA that can be used to reduce the size of the optimization problem. These are orthogonal PCA and orthogonal PLA [5]. Another approach belongs to the three-way method in which a three-way cube of data is decomposed into three loading matrices. These methods are PARAllel FACtors (PARAFAC) and Tucker3 [7]. Furthermore, to analyze multivariate multi-step processes, methods like Priority PLS Regression can be used [8].

Apart from these, there are several more statistical techniques that can be used for multivariate statistical process control (MSPC). These are Structuration des tableaux à trois indices de la statistique (STATIS), multiset canonical correlation analysis (MCCA), multiway independent component analysis (MICA), multiway slow features analysis (MSFA), etc. These techniques come under dimensionality reduction methods. There is one more technique that comes under non-dimensionality reduction methods, i.e., Parallel Coordinates (PARCOORDS). A detailed description of these techniques can be referred to in [9].

2.2.2 Process Monitoring

Batch process monitoring can be performed through a variety of methods. Development of a data-driven system to accurately identify the end of each batch could be one of the process monitoring systems. And to achieve this, techniques such as PLS and MPLS could be applied [10]. Additionally, to monitor and model the multiphase batch process, some other techniques like multiblock PCA, multiblock PLS, sub-PCA, and soft-transition multiple PCA (STMPCA) are used. The STMPCA method can be used to identify and model both process phases and transitions between two phases [11].

To monitor several quality characteristics in a batch process, multivariate control charts (CCs) based on MPCA are generally used. But other quality control strategies for batch process monitoring are also proposed such as the graphical non-parametric strategy based on the STATIS method in case of equal duration batches and its dual version in case of varying duration batches for dimensionality reduction [12].

Process monitoring can also be used to differentiate normal data from faulty ones by considering it as a one-class classification problem. One such one-class classification method for process monitoring is support vector data description (SVDD). It is quite different from traditional data-description methods like PCA/PLS and is also effective for non-linear process modeling. For multiphase and multimode batch processes, the SVDD method is further extended to sub-SVDD which performs better than the sub-PCA method [13]. To further improve the process monitoring performance of the SVDD method, bagging SVDD is proposed which is an ensemble form of the SVDD method [14]. In another research, some data-based batch process monitoring is also proposed which includes functional data description (FDD) and functional SVDD (FSVDD) which is the extended version of the conventional SVDD method [15].

For monitoring key operation units of batch processes, a time-slice canonical correlation analysis (CCA)-based multivariate statistical monitoring is also proposed. To achieve statistical monitoring using this approach, first, the unfolding of three-way batch process data into time-slice data is done, then, to explore the correlation between the key units and the entire process, CCA modeling at each time instant is performed, and finally, generation of fault detection residual and construction of monitoring statistics are done [16].

2.2.3 Fault Detection

For fault detection and classification in complex batch processes, different approaches are proposed. One of the approaches is based on the multiway discrete hidden Markov model (MDHMM). The monitoring results in terms of fault detection and false alarm rates show that MDHMM is superior to conventional MPCA and multiway dynamic principal component analysis (MDPCA) methods [17]. Another research describes the use of warping information for supervised fault classification using different methods such as Partial Least Squares-Discriminant Analysis (PLS-DA) and Soft Independent Modelling of Class Analogy (SIMCA) [18].

For effective fault diagnostics, the use of several algorithms together in stages is also proposed. This approach includes the use of Support Vector Machine (SVM) classifier to detect the abnormal observations followed using K-Means clustering to cluster the normal process dynamics, then, use of PCA to further model each part of the process dynamics, and finally, after projecting the abnormal observations into the PCA models, out of control scenarios are consolidated to extract the fault fingerprints. This approach can detect abnormal observations effectively along with proper fault fingerprints classification [19].

There is also a state-of-the-art data-driven framework for batch process monitoring using a big data approach in which a new feature selection algorithm based on nonlinear Support Vector Machine (SVM) formulations is presented. This new algorithm can be applied for fault detection and diagnosis in batch processes [20]. Detecting minor faults in batch processes through multivariate statistical analysis methods is difficult. Using deep neural networks, such minor faults can be detected as they can extract data features better but due to a large number of connection parameters between layers, they are training-time consuming. To resolve this issue, Broad Learning System (BLS) network structure is proposed as this network structure can be expanded without retraining process

and hence, can save a lot of training time. Moreover, to separate different stages of the production process having different characteristics in a multi-stage batch process, the Affinity Propagation (AP) algorithm is also proposed. With the integration of these two methods, the AP-BLS model is created which has high superiority over other monitoring models in terms of monitoring results [21].

2.2.4 Quality Prediction

To implement an accurate quality prediction for batch processes, several works based on the slow time-varying characteristics are conducted. This includes the ridge regression method based on the batch augmentation analysis framework. As compared to traditional quality prediction methods based on PLS, the results are better in the case of batch augmentation. This state-of-the-art method is described in [22].

2.2.5 Phase Classification

Batch processes generally contain multiple phases and for multi-phase characteristics of such processes, different ways of phase segmentation and modeling strategies are proposed. One of the strategies includes a phase identification method based on different PCA models where every phase is modeled separately based on the phase identification. This method can be used to better capture the process dynamics in different phases [23]. Another research shows that by detecting the phase change events or singular points (SP), batch process monitoring can be achieved, and hence, a moving window based real-time monitoring strategy for SP detection is proposed. It is based on the key hypothesis that around an SP, the statistical properties of the data undergo a significant change [24].

In batch processing, batches generally have unequal durations and are not synchronized. Therefore, methods like dynamic time warping (DTW) are used to synchronize and align the batch durations. Furthermore, to determine the in-control set of batches, three DTW methods are compared using supervised classification through the k-nearest neighbor (KNN) technique in [25]. In another research, a self-adaptive multi-phase batch process fault diagnosis method is proposed for non-linear and multiphase characteristics of batch processes. In this approach, first, a kernel entropy component analysis (KECA) method is used to adaptively achieve multi-phase partition, then, based on these partitioned phases and the effective failure features (extracted through the KECA feature extraction method), a multi-phase KECA failure monitoring model is developed and finally, to automatically recognize each sub-phase fault diagnosis, a multi-phase batch process fault diagnosis method is proposed. This method applies the multi-class support vector machines (MSVM) and fireworks algorithm (FWA). The details of these methods and algorithms can be found in [26]. In separate research, the results show the importance of process phase classification for anomaly detection in a multi-phase industrial process. A two-steps methodology for anomaly detection is proposed: the first step identifies the process phase, and the second step implements the anomaly detection based on the gained information from the first step. In this approach, classification algorithms such as the random forests algorithm (RFA) and decision jungle algorithm (DJA) are used in both steps [27].

Using data-driven modeling for processing data that originates from uneven, multi-phase batches is a challenging task. Therefore, a state-of-the-art machine learning algorithm for non-linear dimensionality reduction is proposed. This algorithm is t-Distributed Stochastic Neighbor Embedding (t-SNE) which can be used to perform unsupervised phase identification via manifold learning [28].

2.3 Change Point Detection

Change point detection (CPD) is the task of finding changes in the time series data. It can also be described as the identification of times when the probability distribution of a stochastic process or time series segment changes. The first works on change point detection were done by E. S. Page in the 1950s for industrial quality control purposes [29] [30]. After that, this method has not only been used in areas of statistics and signal processing but also in various other applications like speech processing, financial analysis, bioinformatics, climatology, network traffic data analysis, etc. Moreover, change point detection can be performed either in real-time or after collecting all the data. When the detection is performed in real-time, it is known as online change point detection and when it is performed after collecting the data, it is known as offline change point detection [1] [3].

Many unsupervised and supervised methods such as probabilistic methods, likelihood ratio, support vector machine (SVM), decision tree, etc. have been proposed to detect change points in time series. A survey on all these

methods can be found in [31]. In another research based on non-parametric divergence estimation between time-series samples from two retrospective segments, a statistical change point detection algorithm is proposed. In this approach, the relative density-ratio estimation method is used for accurate and efficient estimation [32]. Apart from the different methods used for change point detection, proper evaluation of these algorithms is also necessary. A detailed description of these evaluation methods can be found in [33].

For the analysis of batch process data, many data-driven techniques, methods, and algorithms are proposed and applied. Change point detection can also be used for process data analysis as the detected change points in the time series data of process variables may indicate the start or end of process operations, events, or phases. In other words, a change point in process data can be defined as a point in time where an event or a phase in the process starts or ends. Moreover, different unsupervised approaches are proposed for change point detection. One of the proposed approaches is the optimization approach where a cost function is minimized along with a penalty function to obtain the desired change points. Another one is the Bayesian approach where the posterior probability of a specific sample being a change point is calculated based on Bayesian statistics. A detailed description of these approaches for change point detection and their application in process data analysis can be found in [1].

2.4 Existing Challenges

There are several data-driven techniques, methods, and algorithms that can be used for the identification of events in batch process data. But due to the following existing challenges, not all methods are suitable for events identification:

- In a batch process, sometimes the recorded events data are inaccurate and due to this inaccurate labeled data, the results from supervised algorithms are also inaccurate.
- Moreover, sometimes the batch events are not recorded, and it is a very strenuous task to manually label these huge sets of data. Therefore, due to this absence of labeled data, supervised algorithms do not work at all, but the unsupervised algorithm can still work.
- In case of the absence of labels, change point detection based on an unsupervised algorithm can still work but they are largely affected by the choice of cost functions and dynamic, non-linear, non-stationary nature of the process variables. So, the results are still not accurate.
- To increase the accuracy of the results, semi-supervised algorithms can be used in which the user feedback from process experts is incorporated into the algorithm and it will not be a strenuous task because few sets of data can be labeled easily. Therefore, there is a need for a method that can accurately identify batch events by creating artificial labels for them with the help of user feedback from process experts, and interactive change point detection is one of those methods.

In this thesis, first, the unsupervised change point detection method is evaluated for the events identification and its drawbacks are discussed, and then, different approaches are proposed, experimented with, evaluated, and discussed for the interactive change point detection method where the crucial user feedback from process experts is incorporated into the CPD algorithm to increase the accuracy of the events identification results. Moreover, the implementation of a prototype for the interactive process is also discussed.

3 Method

In this chapter, detailed descriptions of different approaches for the events identification are discussed. There are three sections in this chapter: the first section includes the experiments, the evaluations, and the results of the unsupervised change point detection for the events identification, then, different approaches are proposed for the interactive change point detection in the second section, and finally, the implementation of UI prototype for the complete interactive process is described in the third section.

3.1 Unsupervised Change Point Detection

To detect change points in a time series data using unsupervised change point detection, there are two approaches: one is the optimization approach and the other is the Bayesian approach. For the identification of events in process data, an optimization approach for unsupervised change point detection is used for experiments in this thesis. These experiments are not the main part of this thesis, these are performed to prove the need for an interactive change point detection method. There are three components of the optimization approach for change point detection: search method, cost function, and penalty function. In this approach, the purpose is to find the optimum number of change points using a search method along with minimizing the cost function and tuning the penalty function. In general, the search method is used to search all relevant change points in the complete dataset, then, the cost of each of these change points are calculated using a cost function and finally, an initial optimum number of change points are obtained by minimizing this cost function. Moreover, the penalty value is used as a tuning parameter to further optimize the number of detected change points because the number of detected change points is unknown beforehand, and the number of change points decreases with an increase in penalty value [1].

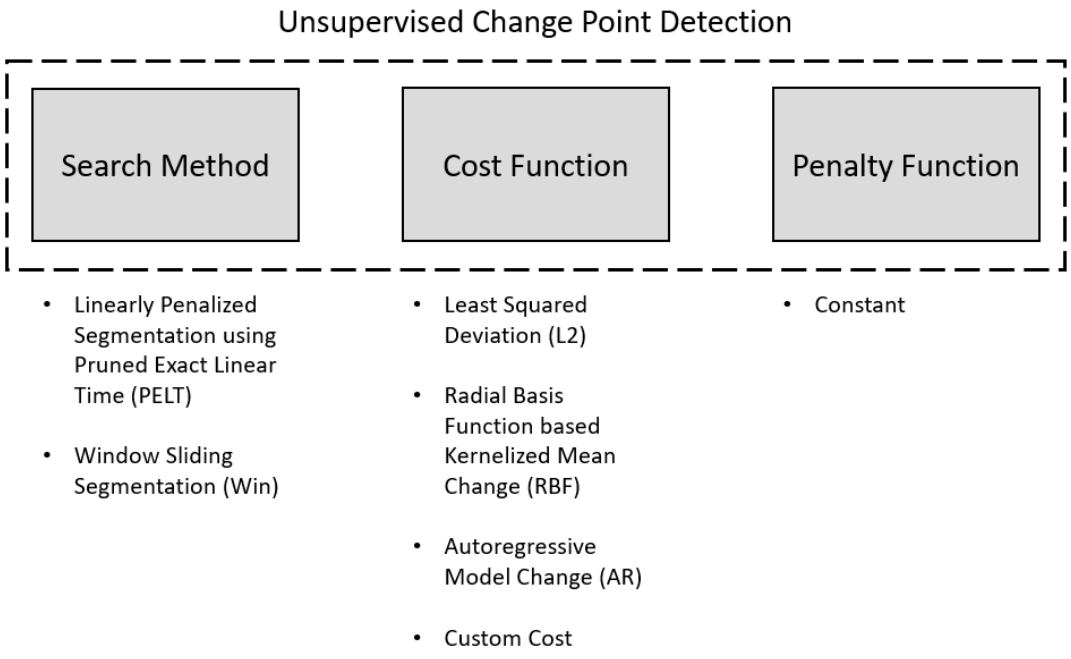

Figure 3.1 Components of optimization approach for unsupervised change point detection [1].

The three components in the optimization approach for unsupervised change point detection, i.e., search method, cost function, and penalty function contain a variety of methods or functions as shown in Figure 3.1. To solve a problem, different combinations of these methods or functions can be used. In this section, various experiments on the combination of some of these methods and functions for events identification are performed, evaluated, and finally, concluded. More specifically, for search method, linearly penalized segmentation using pruned exact linear time (PELT) and window sliding segmentation (Win) is used; for cost function, least squared deviation (L2), radial basis function based kernelized mean change (RBF), autoregressive model change (AR) and other custom costs such as linear regression model change (LinReg), ridge regularization and lasso regularization are used; and finally,

for penalty function, a constant value function is used. All detailed descriptions of these different methods and functions can be found in [1] and [3].

3.1.1 Experiments on Simulated Dataset

The dataset used for experiments with unsupervised change point detection is the simulated dataset. This simulated dataset is generated by developing and simulating a benchmark model of a batch process with full control over disturbances and noises. The details of this simulated dataset can be found in [34].

The simulated dataset is preprocessed before applying the change point detection algorithm on it. Down-sampling and normalizing were the preprocessing steps that were used. The following figures Figure 3.2 and Figure 3.3 illustrate the preprocessed multiple process variables and events from both single batch and multiple batches respectively of the simulated dataset. The colored traces represent the process variables and colored areas represent the multiple events in each batch. In Figure 3.3, five batches of data are shown.

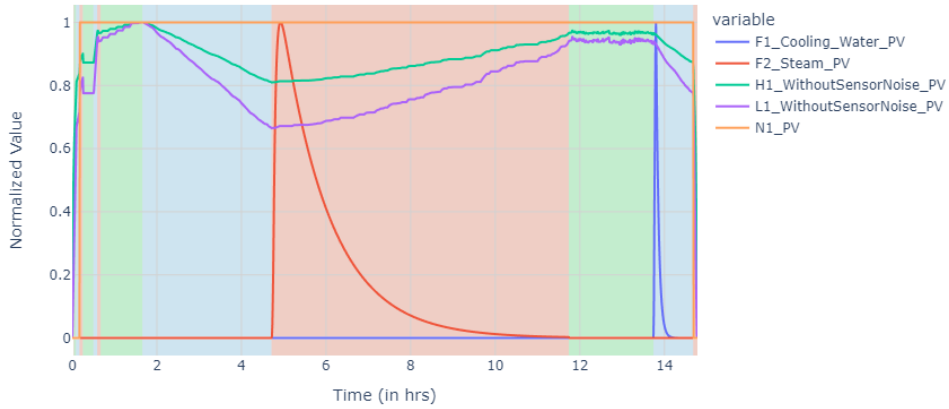

Figure 3.2 Multiple process variables and events from a single batch of the simulated dataset.

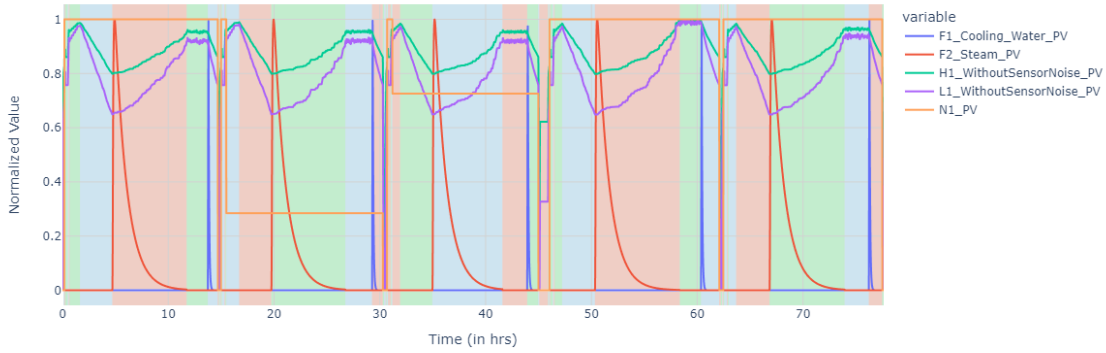

Figure 3.3 Multiple process variables and events from multiple batches of the simulated dataset.

To implement unsupervised change point detection, the python programming language and its libraries can be used. In this thesis, the python library used for performing the experiments with change point detection is ruptures. This library is used for offline change point detection and provides methods for the analysis and segmentation of non-stationary signals. Moreover, it focuses on ease of use by providing a well-documented and consistent interface with a facility to extend the package with different algorithms and models. All other details of this library can be found in [3].

For events identification with unsupervised change point detection, three sets of experiments are performed. The following Table 3.1 describes these three sets of experiments. In this table, the columns represent the name of the

experiment sets, the description of the experiment sets, the dataset used for the experiment sets, the search methods used, the cost functions used, the penalty functions used, and the evaluation metrics/methods used respectively.

Table 3.1 Description of three different sets of experiments performed with unsupervised change point detection for events identification.

S. No.	Experiment Set	Description	Dataset	Search methods	Cost functions	Penalty functions	Evaluation
1.	Experiment U1	To get the unsupervised CPD results for single batch data with different search methods and cost functions by manually optimizing (fixed) penalty value to get close to the optimum or actual number of change points.	Simulated Dataset (single batch data)	• Win • PELT	• l2 • rbf • ar • linReg • ridge • lasso	Constant	• Number of predictions (K) • Annotation error • Meantime • Precision • Recall • F1 score
2.	Experiment U2	To get the unsupervised CPD results for single batch data over a range of penalty values (0.001-10) with different search methods and cost functions.	Simulated Dataset (single batch data)	• Win • PELT	• l2 • linReg • ridge • lasso	Constant	• Detected global change points. • Evaluation by visualizing change points on a range of penalties.
3.	Experiment U3	To get the unsupervised CPD results for multiple batch data over a range of penalty values (0.001-10) with different search methods and cost functions.	Simulated Dataset (multiple batch data)	Win	• l2 • linReg • ridge • lasso	Constant	• Detected global change points. • Evaluation by visualizing change points on a range of penalties.

Each of these three sets of experiments contains several experiments composed by taking different combinations of search methods and cost functions. The detailed descriptions of all these experiments are following:

Experiment U1

The aim of this first set of experiments is to get unsupervised CPD results for single batch data with different search methods and cost functions by manually optimizing (fixed) penalty value to get close to the optimum or actual number of change points. The dataset used for these experiments is the simulated dataset containing a single batch of data. Moreover, for search methods, Win and PELT are used; for cost functions, l2, rbf, ar, linReg, ridge, and lasso are used; for penalty function, a constant value function is used. Along with these, different evaluation metrics are also used for evaluating the results. These evaluation metrics are Number of predictions (K), Annotation error, Meantime, Precision, Recall, and F1 score. All detailed descriptions of these evaluation metrics can be found in [1] and [3].

The number of experiments performed under this first set of experiments is 12. The following Table 3.2 describes these 12 experiments. In this table, the columns represent the name of the experiments, the dataset with the number of batches used for the experiments, the search methods used, the cost functions used, and the penalty values used respectively.

Table 3.2 Description of experiments under the first set of experiments.

S. No.	Experiment	Dataset	Search method	Cost function	Penalty value
1.	Experiment U1.1	Simulated Dataset (1 batch)	Win (window width: 10)	l2	0.001

2.	Experiment U1.2	Simulated Dataset (1 batch)	Win (window width: 10)	rbf	0.1
3.	Experiment U1.3	Simulated Dataset (1 batch)	Win (window width: 10)	ar	0.0001
4.	Experiment U1.4	Simulated Dataset (1 batch)	Win (window width: 10)	linReg	1
5.	Experiment U1.5	Simulated Dataset (1 batch)	Win (window width: 10)	ridge	1
6.	Experiment U1.6	Simulated Dataset (1 batch)	Win (window width: 10)	lasso	1
7.	Experiment U1.7	Simulated Dataset (1 batch)	PELT	l2	1
8.	Experiment U1.8	Simulated Dataset (1 batch)	PELT	rbf	30
9.	Experiment U1.9	Simulated Dataset (1 batch)	PELT	ar	0.001
10.	Experiment U1.10	Simulated Dataset (1 batch)	PELT	linReg	10
11.	Experiment U1.11	Simulated Dataset (1 batch)	PELT	ridge	1
12.	Experiment U1.12	Simulated Dataset (1 batch)	PELT	lasso	10

Experiment U2

The aim of the second set of experiments is to get unsupervised CPD results for single batch data over a range of penalty values (0.001-10) with different search methods and cost functions. The dataset used for these experiments is the simulated dataset containing a single batch of data. Moreover, for search methods, Win and PELT are used; for cost functions, l2, linReg, ridge, and lasso are used; for penalty function, a constant value function is used. Along with these, the number of global change points and visualizing change points on a range of penalties are also used for evaluating the results. Global change points are all the change points that occur in the complete range of penalties.

The number of experiments performed under the second set of experiments is 5. The following Table 3.3 describes these 5 experiments. In this table, the columns represent the name of the experiments, the dataset with the number of batches used for the experiments, the search methods used, the cost functions used, and the penalty range used respectively.

Table 3.3 Description of experiments under the second set of experiments.

S. No.	Experiment	Dataset	Search method	Cost function	Penalty range
1.	Experiment U2.1	Simulated Dataset (1 batch)	Win (window width: 10)	l2	(0.001-10)
2.	Experiment U2.2	Simulated Dataset (1 batch)	Win (window width: 10)	linReg	(0.001-10)
3.	Experiment U2.3	Simulated Dataset (1 batch)	Win (window width: 10)	ridge	(0.001-10)
4.	Experiment U2.4	Simulated Dataset (1 batch)	Win (window width: 10)	lasso	(0.001-10)

5.	Experiment U2.5	Simulated Dataset (1 batch)	PELT	l2	(0.001-10)

Experiment U3
The aim of the second set of experiments is to get unsupervised CPD results for multiple batch data over a range of penalty values (0.001-10) with different search methods and cost functions. The dataset used for these experiments is the simulated dataset containing multiple batches of data. Moreover, for the search method, Win is used; for cost functions, l2, linReg, ridge, and lasso are used; for penalty function, a constant value function is used. Along with these, the number of global change points and visualizing change points on a range of penalties are also used for evaluating the results. Global change points are all the change points that occur in the complete range of penalties.

The number of experiments performed under the third set of experiments is 4. The following Table 3.4 describes these 4 experiments. In this table, the columns represent the name of the experiments, the dataset with the number of batches used for the experiments, the search methods used, the cost functions used, and the penalty range used respectively.

Table 3.4 Description of experiments under the third set of experiments.

S. No.	Experiment	Dataset	Search method	Cost function	Penalty range
1.	Experiment U3.1	Simulated Dataset (5 batches)	Win (window width: 10)	l2	(0.001-10)
2.	Experiment U3.2	Simulated Dataset (5 batches)	Win (window width: 10)	linReg	(0.001-10)
3.	Experiment U3.3	Simulated Dataset (5 batches)	Win (window width: 10)	ridge	(0.001-10)
4.	Experiment U3.4	Simulated Dataset (5 batches)	Win (window width: 10)	lasso	(0.001-10)

3.1.2 Results

There are three sets of experiments that are performed with unsupervised change point detection for the identification of the events. The descriptions of all these three sets of experiments are described in the experiments on simulated dataset sub-section 3.1.1 and the results are presented in this sub-section.

Results of Experiment U1
There are 12 experiments in the first set of experiments. The results of Experiment U1.1 and Experiment U1.7 are shown in the following figures Figure 3.4 and Figure 3.5 respectively. Rest of the results are demonstrated in Appendix A. These figures illustrate multiple process variables, events, and detected change points for a single batch from the simulated dataset using unsupervised CPD. The colored traces represent the process variables, colored areas represent the events and vertical dashed lines represent the detected change points. From these figures, it can be observed that some detected change points are overlapped with the start and the end of some events, but most of them are not overlapping, so, it can be said that the overall result is not good.

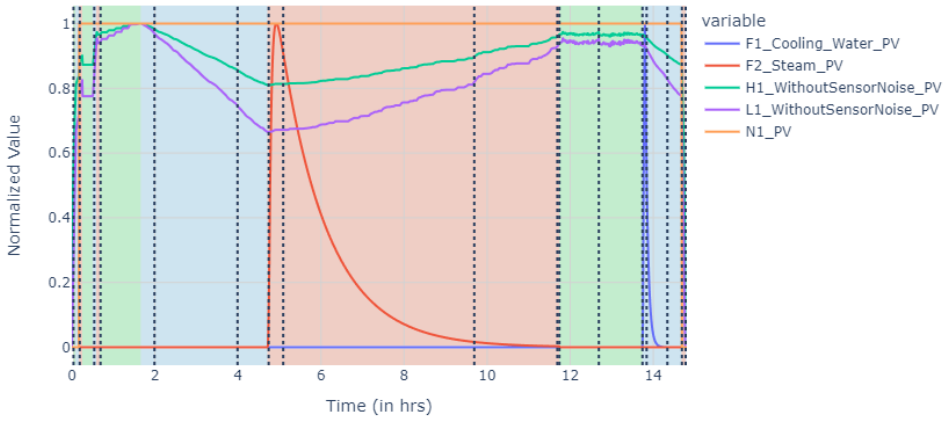

Figure 3.4 Process variables, events, and detected change points obtained from Experiment U1.1

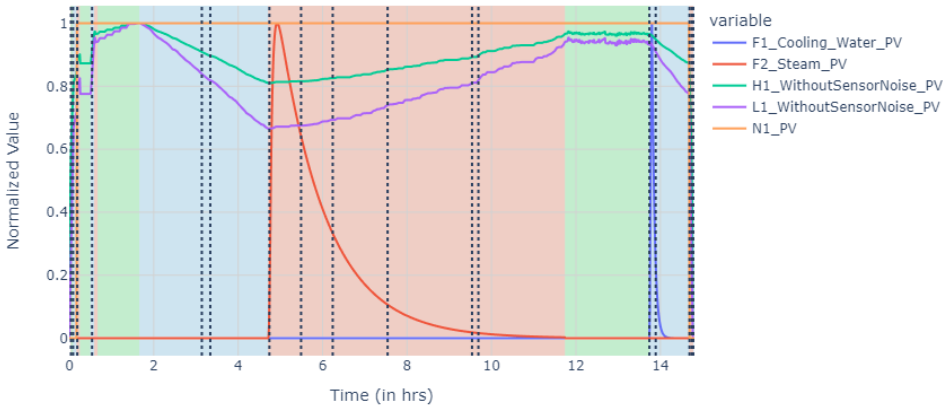

Figure 3.5 Process variables, events, and detected change points obtained from Experiment U1.7

The following Table 3.5 demonstrates the summary of results from the first set of experiments. In this table, the columns represent the name of the experiments, the search methods used, the cost functions used, the penalty values used, the values from the used evaluation metrics, and the time taken to compute the results respectively.

Table 3.5 Summary of results from the first set of experiments.

Experiment	Search Method	Cost Function	Penalty	Number of Predictions (K)	Annotation Error	Meantime	Precision	Recall	F1	Computation Time
Experiment U1.1	WIN (width = 10)	l2	0.001	17	4	0.315	0.125	0.167	0.143	2.8s
Experiment U1.2		rbf	0.1	14	1	0.312	0.154	0.167	0.16	1.2s
Experiment U1.3		ar	0.0001	11	2	0.234	0.2	0.167	0.182	0.6s
Experiment U1.4		linReg	1	16	3	0.525	0.133	0.167	0.148	0.5s
Experiment U1.5		ridge	1	8	5	0.017	0.286	0.167	0.211	5.3s
Experiment U1.6		lasso	1	9	4	0.067	0.25	0.167	0.2	3.2s
Experiment U1.7	PELT	l2	1	17	4	0.784	0.188	0.25	0.214	10.6s
Experiment U1.8		rbf	30	13	0	1.48	0.333	0.333	0.333	41.2s
Experiment U1.9		ar	0.001	12	1	0.027	0.182	0.167	0.174	15.6s
Experiment U1.10		linReg	10	16	3	0.994	0.2	0.25	0.222	6.8s
Experiment U1.11		ridge	1	12	1	1.054	0.273	0.25	0.261	1m 42.4s
Experiment U1.12		lasso	10	10	3	1.448	0.333	0.25	0.286	3m 10.6s

The results from the first set of experiments show that the unsupervised CPD algorithm for events identification has not performed well. Despite tuning the different parameters, the desired results are not obtained as most of the detected change points do not overlap with the start and the end of the events.

Results of Experiment U2

There are 5 experiments in the second set of experiments. The results of Experiment U2.1 and Experiment U2.5 are shown in the following figures Figure 3.6Figure 3.4 and Figure 3.7Figure 3.5 respectively. Rest of the results are demonstrated in Appendix B. These figures illustrate multiple process variables, events, and detected change points over a range of penalties for a single batch from the simulated dataset using unsupervised CPD. The colored traces represent the process variables, the colored areas represent the events, and the vertical solid lines represent the detected change points. The height of the lines indicates the range of penalty values in which that detected change point exists. From these figures, it can be observed that some detected change points are overlapped with the start and the end of some events, but most of them are not overlapping, so, it can be said that the overall result is not good. Moreover, the ranges of penalties for different detected change points are not same.

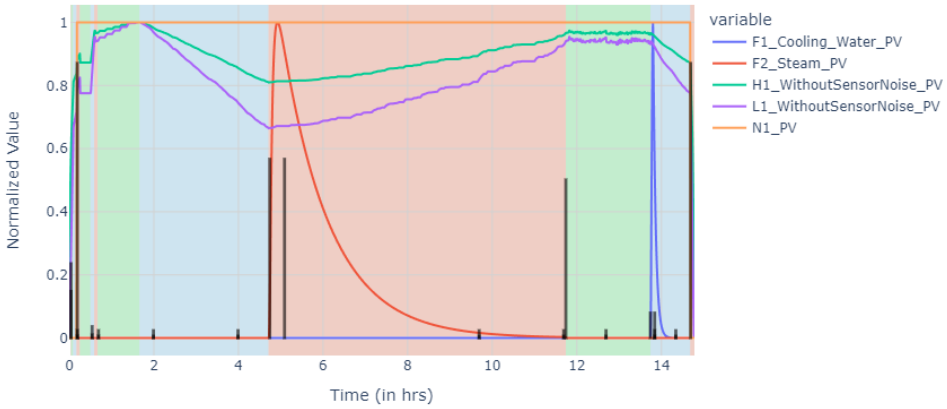

Figure 3.6 Process variables, events, and detected change points obtained from Experiment U2.1

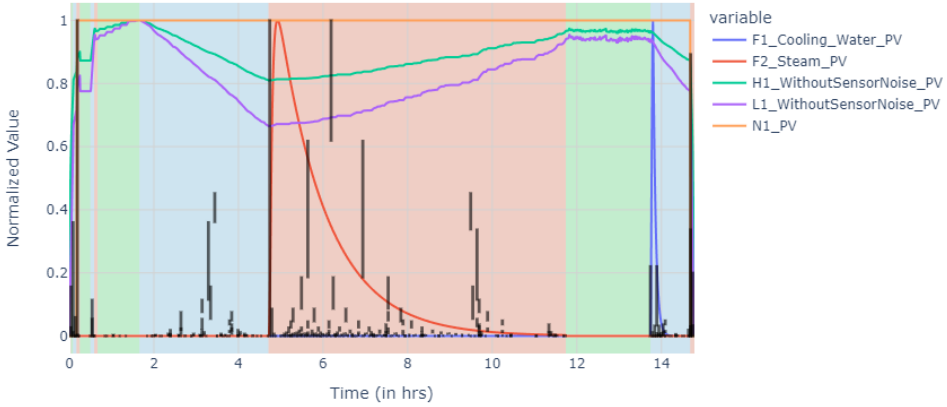

Figure 3.7 Process variables, events, and detected change points obtained from Experiment U2.5

The following Table 3.6 demonstrates the summary of results from the second set of experiments. In this table, the columns represent the name of the experiments, the search methods used, the cost functions used, the penalty range used, the values from the used evaluation metric, and the time taken to compute the results respectively.

Table 3.6 Summary of results from the second set of experiments.

Experiment	Search Method	Cost Function	Penalty Range	Detected Global Change Points	Computation Time
Experiment U2.1	WIN (width = 10)	l2	(0.001-10)	16	1.1s
Experiment U2.2		linReg	(0.001-10)	32	6.8s
Experiment U2.3		ridge	(0.001-10)	29	7.2s
Experiment U2.4		lasso	(0.001-10)	8	5.6s
Experiment U2.5	PELT	l2	(0.001-10)	169	7m 23.2s

The results from the second set of experiments show that the unsupervised CPD algorithm for events identification has not performed well. The desired results are not obtained as most of the detected change points do not overlap with the start and the end of the events.

Results of Experiment U3

There are 4 experiments in the third set of experiments. The result of Experiment U3.1 is shown in the following Figure 3.8. Rest of the results are demonstrated in Appendix C. This figure illustrates multiple process variables, events, and detected change points over a range of penalties for multiple batches from the simulated dataset using unsupervised CPD. The colored traces represent the process variables, the colored areas represent the events, and the vertical solid lines represent the detected change points. The height of the lines indicates the range of penalty values in which that detected change point exists. From this figure, it can be observed that some detected change points are overlapped with the start and the end of some events, but most of them are not overlapping, so, it can be said that the overall result is not good. Moreover, the range of penalties for the detected change points changes for different batches. Therefore, the change points obtained are not consistent in all the batches for a fixed penalty value.

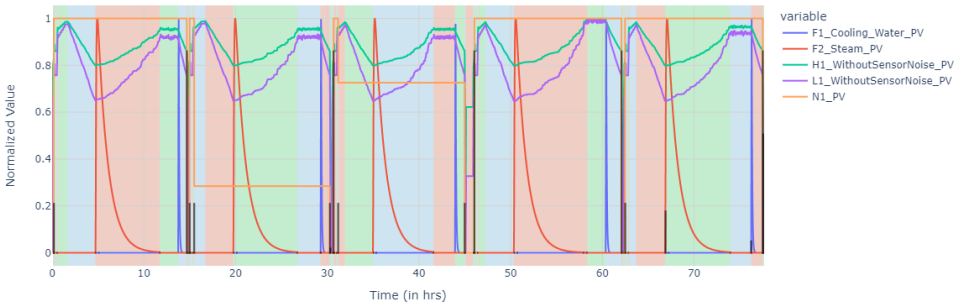

Figure 3.8 Process variables, events, and detected change points obtained from Experiment U3.1

The following Table 3.7 demonstrates the summary of results from the third set of experiments. In this table, the columns represent the name of the experiments, the search methods used, the cost functions used, the penalty range used, the values from the used evaluation metric, and the time taken to compute the results respectively.

Table 3.7 Summary of results from the third set of experiments.

Experiment	Search Method	Cost Function	Penalty Range	Detected Global Change Points	Computation Time
Experiment U3.1	WIN (width = 10)	l2	(0.001-10)	46	2.3s
Experiment U3.2		linReg	(0.001-10)	147	1m 24.3s
Experiment U3.3		ridge	(0.001-10)	84	25.3s
Experiment U3.4		lasso	(0.001-10)	32	20.8s

The results from the third set of experiments show that the unsupervised CPD algorithm for events identification has performed badly for multiple batches. The desired results are not obtained as most of the detected change

points do not overlap with the start and the end of the events. Moreover, the results have further deteriorated for multiple batches of data due to inconsistent detected change points in different batches for a fixed penalty value.

3.1.3 Findings and Conclusion

The following are some findings from the experiments with unsupervised CPD for identifications of events:

- Cost functions ridge and lasso with WIN search method are performing comparatively fine with the lowest meantime evaluation metric.
- Cost function linReg produces more change points as compared to ridge and lasso for the same penalty value.
- Cost function rbf requires a high penalty value to reach close to an optimum number of change points.
- Cost function ar requires a low penalty value to reach close to an optimum number of change points.
- In general, the PELT search method takes a longer computational time than the WIN search method.
- Some detected change points are overlapped with the start and the end of some events, but most of them are not overlapping, so, it can be said that the overall result is not good.
- The ranges of penalties for different detected change points are not same.
- For multiple batches of data, the range of penalties for the detected change points changes for different batches. Therefore, the change points obtained are not consistent in all the batches for a fixed penalty value.

The following conclusions are drawn from the above findings:

- The results from unsupervised change point detection largely depend on the choice of search methods, cost functions, and penalty values.
- Despite tuning the different parameters, the desired results were not achieved due to the dynamic, nonlinear, and non-stationary nature of the process variables.
- Moreover, results have further deteriorated for multiple batches of data.
- Due to these drawbacks of unsupervised CPD, there is a need for another approach. Therefore, a new approach is proposed in this thesis, i.e., interactive change point detection algorithm.

In the next section 3.2, this interactive change point detection approach is discussed.

3.2 Interactive Change Point Detection

To overcome the issues to identify events in the batch process using unsupervised change point detection, a new semi-supervised approach is proposed in this thesis. This new approach is interactive change point detection. In this approach, user feedback is taken into account for generating both initial and updated results. Since the ground truth events may or may not be available, therefore, the interactive CPD approach does not necessarily depend on ground truth events rather it depends on the feedback from the process experts. The process experts are the users with the knowledge of the process variables and the events. However, the process experts may optionally decide to take help from the available ground truth events, but it is not necessary. Therefore, the interactive CPD approach will work irrespective of the availability of the ground truth events. Furthermore, three components of interactive change point detection are proposed in this thesis. These three components are search methods, customized cost functions, and tuning parameters where the user feedback is a part of two components, i.e., customized cost functions and tuning parameters. This approach is called interactive because the cost is calculated as well as the parameters are tuned based on user feedback. In unsupervised CPD, identification of all the events was performed at the same time but in the interactive CPD approach, identification of one event across multiple batches at a time is performed and then, the complete process is repeated for all the other events. In the interactive CPD approach, the same event in different batches is considered to contain similar segments and since the users are the process experts, they can select a segment of their interest based on their knowledge of the events. After obtaining the user-selected segment, the purpose is to find the segments similar to the user-selected segment which are present in the complete dataset using a search method along with minimizing the cost function and tuning different parameters. In general, the search method is used to search all relevant change points in the complete dataset, then, the cost of each of the segments obtained from all these change points are calculated based on user-selected segment using a customized cost function and finally, an initial optimum number of change points are obtained by minimizing this cost function. These initially optimized change points can then be processed to find the segments and generate artificial labels for the event as the initial results. Moreover, different tuning parameters are used to further optimize

the number of detected change points with the help of evaluation and user feedback and hence can generate more accurate artificial labels for the event as the updated results.

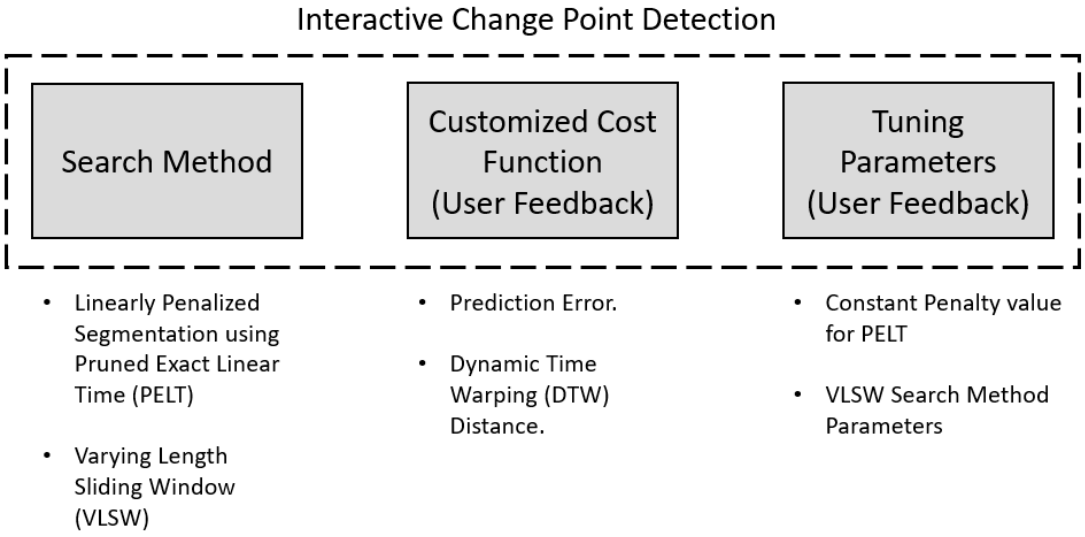

Figure 3.9 Components of interactive change point detection.

The three components in the interactive change point detection, i.e., search method, customized cost function, and tuning parameters contain a variety of methods or functions as shown in Figure 3.9. More specifically, for search method, linearly penalized segmentation using pruned exact linear time (PELT) and varying length sliding window (VLSW) are used; for customized cost function, prediction error and dynamic time warping (DTW) distance are used; and finally, for tuning parameters, a constant value penalty function for PELT and VLSW search method parameters are used. To achieve the identification of events in the batch process data using interactive change point detection, the complete workflow, experiments with different combinations of these methods or functions on simulated as well as the real-life dataset, and the implementation of a prototype are discussed, performed, evaluated and finally, concluded in this thesis. Since, the interactive CPD approach works irrespective of the availability of the ground truth events, therefore, to show this, the examples for the identification of the events using the interactive CPD approach through a UI prototype are shown without the ground truth events in section 3.3. However, in the experiments with interactive CPD approach for identification of the events in sections 4.1 and 4.2, ground truth events are only used for evaluating the results through different evaluation metrics but there is also one evaluation method that can be used for evaluating the results in case of absence of the ground truth events. In this section, first, the basic terminologies used in this thesis are explained and then, the complete workflow along with different methods used for the interactive CPD are thoroughly discussed.

3.2.1 Terminologies Used

The following points explain some basic terminologies that are used throughout this thesis:

- A batch is one run of a recipe that is used for the production of materials.
- A recipe is a sequence of operations to be executed for producing certain materials.
- A batch process is a chemical process that produces materials in batches.
- A process variable is a physical quantity associated with a batch process.
- A signal is a time series dataset comprising measurement of one process variable from single or multiple batches.
- A sample contains the values of one or more signals at a particular timestamp.
- A segment contains the samples from consecutive timestamps. The start and end timestamps of these segments can either be given by the user or created by an algorithm, which in this thesis is a search method based on varying length sliding window (VLSW).
- A window contains the start and end timestamps and is used to create segments.
- In this thesis, a sliding window is a window that moves from start to end of the signal.
- The length of the segment is defined as the number of samples in the segment.

3.2.2 Workflow

The complete workflow of the interactive change point detection approach is discussed here. In this approach, the following steps are followed:

- First, the time series dataset is preprocessed.

- After preprocessing steps, users select a single segment of their interest from the visual representation of time series process data through a user interface (UI) prototype based on their knowledge of events and the characteristics of the process variables irrespective of the availability of ground truth events.

- Then, a customized cost function is used where the cost is calculated based on user feedback. In this thesis, the following two customized cost functions are used:
  1. Prediction error from a trained regression model based on the user-selected segment.
  2. Dynamic time warping (DTW) distance from the user-selected segment.

- After that, a search method is used. In this thesis, the following two search methods are used:
  1. Linearly penalized segmentation using the pruned exact linear time (PELT): This method is already explained and discussed in [1] [3].
  2. Varying length sliding window (VLSW): The algorithm of this new method is developed, implemented, and explained in this thesis.

- Then, the customized cost function and the search method are used together for detecting change points. The search method is used to search all relevant change points in the complete dataset, then, the cost of each of the segments obtained from all these change points are calculated based on user-selected segment using a customized cost function and finally, an initial optimum number of change points are obtained by minimizing this cost function. And after postprocessing of these detected change points, the artificial labels for the events are generated as an initial result.

- After getting the initial results, an evaluation of the results is performed. With the help of evaluation, parameter tuning is done through user feedback to further update and increase the accuracy of the results as well as get the updated artificial labels for the events.

The following Figure 3.10 shows the complete workflow of the interactive change point detection approach:

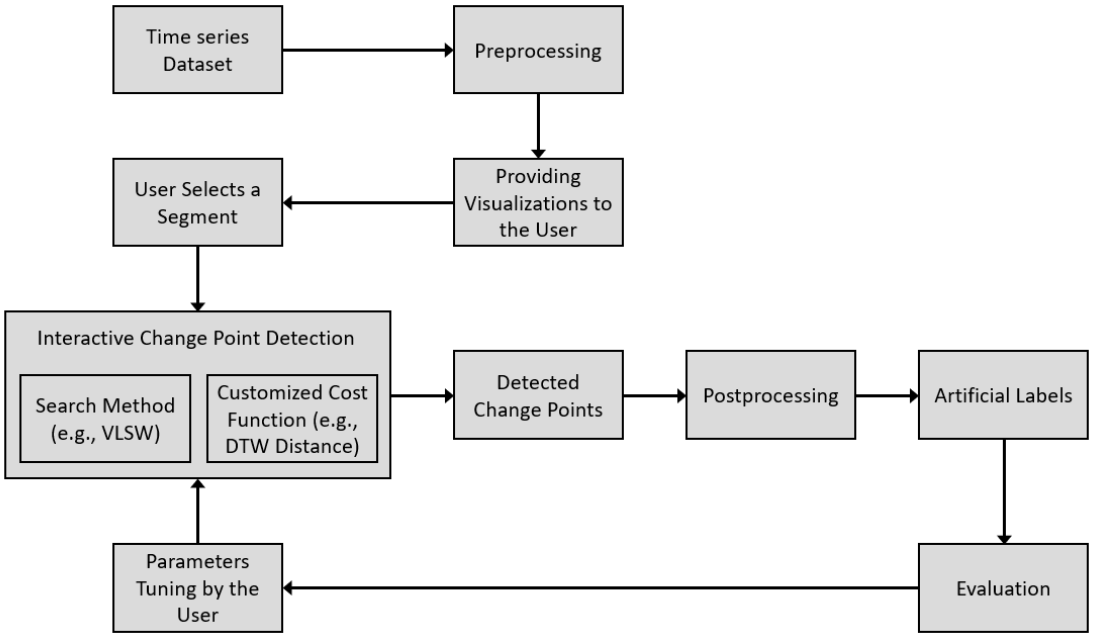

Figure 3.10 Complete workflow of interactive change point detection algorithm.

3.2.3 Preprocessing

The recorded time series dataset from process industries generally contains raw data. These raw data may contain missing values or outliers. Therefore, these raw data need to be cleaned before applying any algorithm or method to analyze them. To clean the raw data, preprocessing is used. Some preprocessing steps are filtering out not-a-number (NaN) values, filling missing values, equidistance sampling, down-sampling, normalization, etc. Some of these preprocessing steps are used in the experiments sections 4.1 and 4.2.

3.2.4 Segment Selection

For identification of events in process data or to generate artificial labels for it, user feedback can play an important role as the users are process experts and can help to generate initial as well as updated results with more accuracy based on their knowledge of events and characteristics of the process variables irrespective of the availability of ground truth events. Therefore, in the interactive CPD approach, users can select a single segment of their interest from the visual representation of time series process data through a user interface (UI) prototype and this user-selected segment is used as a reference to calculate the cost for detecting change points. The detailed discussion of segment selection by the user through a UI prototype is discussed in section 3.3.

3.2.5 Customized Cost Functions

The cost functions in change point detection are used to optimize the result. The optimum change points are obtained by minimizing the cost function. In interactive CPD, the customized cost functions are the cost functions where the cost is calculated based on user feedback. This means that the user-selected segment is used as a reference to calculate the cost for detecting change points. The two customized cost functions used in this thesis are following:

1. Prediction error as a cost function.
2. Dynamic Time Warping (DTW) distance as a cost function.

Prediction error as a cost function
In this type of customized cost function, the cost is calculated as a prediction error from a trained regression model based on the user-selected segment. This means that a regression model is trained on the user-selected segment and then, the cost is calculated as the prediction error from this trained model on all the segments obtained from the search method.

Moreover, the following two types of trained models can be used based on the characteristics of the process variable contained in the user-selected segment:

- Linear Model: If the process variable contained in the user-selected segment has linear characteristics, then, a linear regression model can be used.

- Non-linear Model: If the process variable contained in the user-selected segment has non-linear characteristics, then, a polynomial regression model can be used.

DTW Distance as a cost function
Dynamic time warping (DTW) distance gives a similarity measure between two time series data. These time series data can have different sizes, but they should not have different dimensions. DTW distance can be calculated for both univariate and multivariate time series data. More details about DTW distance can be found in [35] [36]. To calculate DTW distance, the tslearn python library is used throughout this thesis [35].

DTW distance can be used be as a customized cost function where the cost is calculated as a DTW distance between the user-selected segment and all the segments obtained from the search method. The user-selected segment can contain a single process variable (univariate data) or multiple process variables (multivariate data).

3.2.6 Search Methods

The search methods in change point detection are used to search all relevant change points in the complete dataset. The two search methods used in the experiments with the interactive CPD approach are following:

1. Linearly penalized segmentation using the pruned exact linear time (PELT)

2. Varying length sliding window (VLSW)

Linearly penalized segmentation using the pruned exact linear time (PELT)
In this search method, the algorithm uses a pruning rule to discard the partitions while retaining the ability to find the optimal segmentation. Moreover, the penalty value can be tuned to further improve the results obtained by using the PELT search method. More details of this search method can be found in [1] [3]. To use this search method, the ruptures python library is used throughout this thesis. This library is used for offline change point detection and provides methods for the analysis and segmentation of non-stationary signals. Moreover, it focuses on ease of use by providing a well-documented and consistent interface with a facility to extend the package with different algorithms and models. All other details of this library can be found in [3].

Varying length sliding window (VLSW)
The algorithm for this new search method is developed, implemented, and explained in this thesis. In the interactive CPD approach, the same event in different batches is considered to contain similar segments and since the users are the process experts, they can select a segment of their interest based on their knowledge of the events. After obtaining the user-selected segment, the purpose is to find the segments similar to the user-selected segment which are present in the complete dataset using a search method along with minimizing the cost function and tuning different parameters. Therefore, there is a need for a search method that can accurately find those segments that are similar to the user-selected segment and can return the change points obtained from those found segments in the complete dataset. Hence, a new search method, i.e., the varying length sliding window (VLSW) search method is proposed in this thesis. This new varying length sliding window (VLSW) search method aims to answer the first research question of this thesis, i.e., how to achieve the identification of the events for process data through interactive CPD. In the varying length sliding window (VLSW) search method, a sliding window with varying length creates different segments called sliding window segments while moving through the complete time series dataset so that the generated segments cannot have both the same starting index and length. The length of the sliding window segments varies from a minimum length to a maximum length with steps of some samples depending on the steps parameter. The minimum and the maximum length of the sliding window segments depend on the length of the user-selected segment as well as the minimum and the maximum deviation parameters respectively. For the detailed explanation of the VLSW search method along with mathematical representation, a simulated dataset is taken here as an example [34]. This dataset is down-sampled and normalized before applying the VLSW search method on it. The following Figure 3.11 illustrates the preprocessed process variable and the ground truth event from multiple batches of the simulated dataset. In this figure, five batches of data are shown where the blue-colored trace represents the process variable, and the green-colored areas represent the same ground truth event in all these five batches.

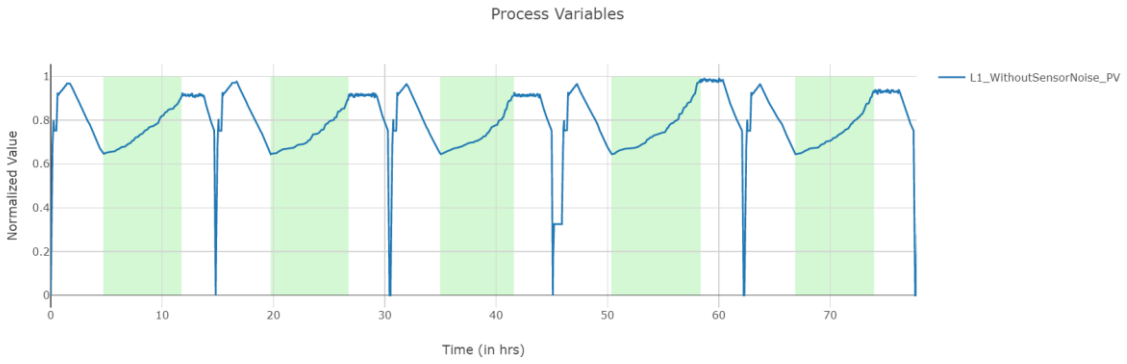

Figure 3.11 Process variable and the ground truth event from multiple batches of the simulated dataset.

The following Figure 3.12 illustrates the user-selected segment, the process variable, and the ground truth event from multiple batches of the simulated dataset. In this figure, five batches of data are shown where the light blue-colored area with the blue-colored border represents the user-selected segment, the blue-colored trace represents the process variable, and the green-colored areas represent the same ground truth event in all these five batches. Let the length of the user-selected segment, the steps parameter, the minimum, and the maximum deviation parameters be denoted by N, r, d_{min} and d_{max} respectively. Then,

$$\text{Minimum length of the sliding window segments} = (N - d_{\min}) \tag{3.1}$$

$$\text{Maximum length of the sliding window segments} = (N + d_{max}) \tag{3.2}$$

Since, from the above equations 3.1 and 3.2, the minimum and the maximum length of the sliding window segments are given as $(N - d_{\min})$ and $(N + d_{max})$ respectively. Therefore, the length of the sliding window segments will vary from $(N - d_{\min})$ to $(N + d_{max})$ with steps of r samples. For example, if $N = 100$, $r = 5$, $d_{\min} = d_{\max} = 20\% \, of \, N$, then, the length of the sliding window segments will range from 80 to 120 with steps of 5 samples.

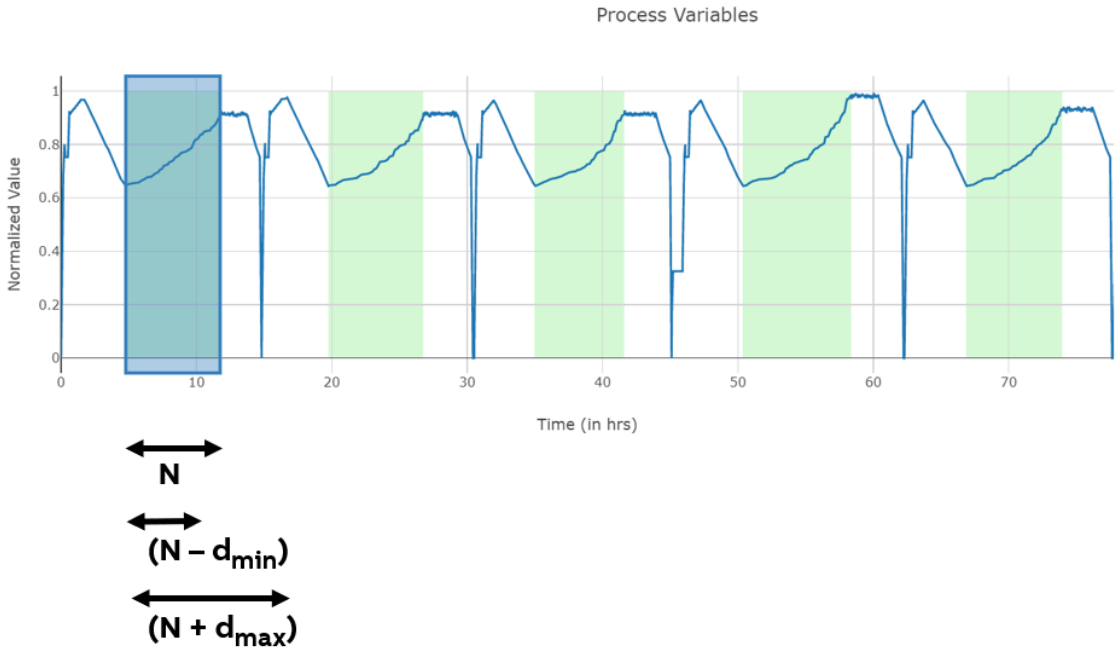

Figure 3.12 User-selected segment, the process variable, and the ground truth event from multiple batches of the simulated dataset.

The following Figure 3.13 illustrates the varying length sliding window, the process variable, and the ground truth event from multiple batches of the simulated dataset. In this figure, five batches of data are shown where the rectangular areas with the solid and dashed borders represent the sliding window at the same starting index but at different lengths, the small dashed arrow represents the variation in the length of the sliding window, the large dashed arrow represents the movement of the sliding window through the complete dataset, blue-colored trace represents the process variable, and the green-colored areas represent the same ground truth event in all these five batches. For creating sliding window segments, the sliding window will start with its starting index at the first index of the complete dataset and its length equal to the minimum length of the sliding window segments, i.e., $(N - d_{\min})$ and will create the first sliding window segment. Then, the sliding window will vary its length from the minimum length of the sliding window segments, i.e., $(N - d_{\min})$ to the maximum length of the sliding window segments, i.e., $(N + d_{\max})$ with steps of r samples while maintaining its starting index position and will create different sliding window segments with different lengths. After this, the sliding window will move to the next index of the dataset with steps of r samples and again vary its length from $(N - d_{\min})$ to $(N + d_{\max})$ with steps of r samples while maintaining its starting index position and will again create different sliding window segments with different lengths. It will continue to do so until it reaches the end of the complete dataset. In this way, many sliding window segments will be created in such a way that no two sliding window segments will have both the same starting index and length. All these generated sliding window segments will then be compared with the user-selected segment to find the segments that are similar to the user-selected segment. Finally, the change points obtained from those found segments in the complete dataset will be returned.

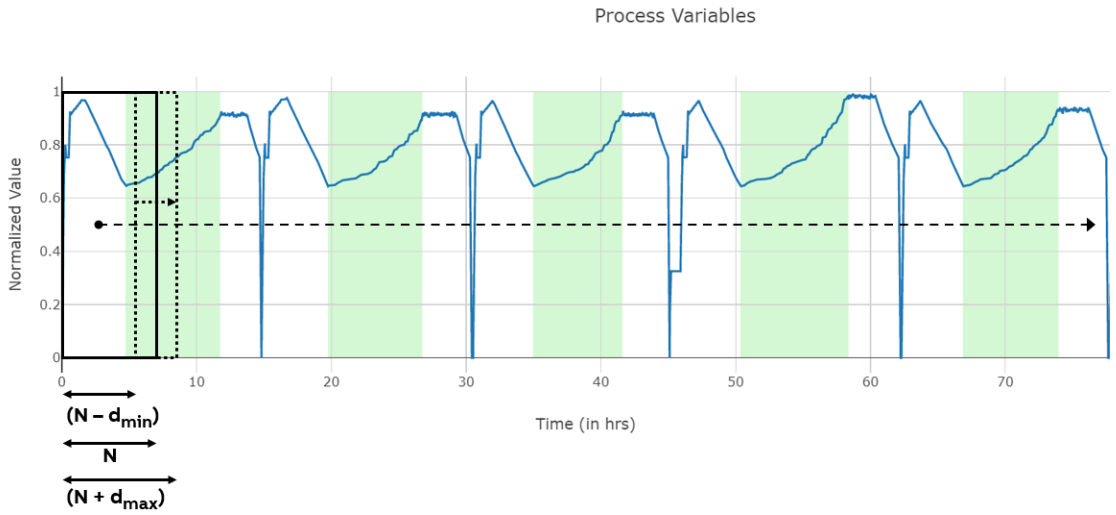

Figure 3.13 Varying length sliding window, the process variable, and the ground truth event from multiple batches of the simulated dataset.

Let a vector describing the complete time series data containing one signal with S number of samples be,

$$\boldsymbol{y} = [y_{t_0}, y_{t_1}, y_{t_2}, \dots, y_{t_{S-1}}] \tag{3.3}$$

In the above equation 3.3, $\boldsymbol{y}$ is a vector that contains a time series signal; t_0 and t_{S-1} are the start and end timestamps of the complete dataset respectively; y_{t_0}, y_{t_1}, y_{t_2} and $y_{t_{S-1}}$ are values of that signal at timestamps t_0, t_1, t_2 and t_{S-1} respectively.

Let a matrix describing the complete time series data containing H multiple signals each with S number of samples be,

$$Y = Y_{t_0,\dots,t_{S-1}} = \begin{bmatrix} \boldsymbol{y}_0 \\ \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_{H-1} \end{bmatrix} = \begin{bmatrix} y_{0,t_0} & y_{0,t_1} & y_{0,t_2} & \cdots & y_{0,t_{S-1}} \\ y_{1,t_0} & y_{1,t_1} & y_{1,t_2} & \cdots & y_{1,t_{S-1}} \\ y_{2,t_0} & y_{2,t_1} & y_{2,t_2} & \cdots & y_{2,t_{S-1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{H-1,t_0} & y_{H-1,t_1} & y_{H-1,t_2} & \cdots & y_{H-1,t_{S-1}} \end{bmatrix}_{S\times H} \tag{3.4}$$

In the above equation 3.4, $\boldsymbol{y}_0$, $\boldsymbol{y}_1$, $\boldsymbol{y}_2$ and $\boldsymbol{y}_{H-1}$ are the vectors that contain one time series signal each and Y is a matrix that contains these H multiple vectors; t_0 and t_{S-1} are the start and end timestamps of the complete dataset respectively; y_{0,t_0}, y_{0,t_1}, y_{0,t_2} and $y_{0,t_{S-1}}$ are values of the first signal (represented by the vector $\boldsymbol{y}_0$) at timestamps t_0, t_1, t_2 and t_{S-1} respectively; y_{H-1,t_0}, y_{H-1,t_1}, y_{H-1,t_2} and $y_{H-1,t_{S-1}}$ are values of the last signal (represented by the vector $\boldsymbol{y}_{H-1}$) at timestamps t_0, t_1, t_2 and t_{S-1} respectively; Y_{t_0} and $Y_{t_{S-1}}$ are the samples of the time series dataset at timestamps t_0 and t_{S-1} respectively; $Y_{t_0,\dots,t_{S-1}}$ represents the segment of the time series dataset that contains the samples of consecutive timestamps from start timestamp t_0 to end timestamp t_{S-1}.

Let the user-selected segment be,

$$Y_{ref} = Y_{t_k,\dots,t_{k+N-1}} = \begin{bmatrix} y_{0,t_k} & y_{0,t_{k+1}} & y_{0,t_{k+2}} & \cdots & y_{0,t_{k+N-1}} \\ y_{1,t_k} & y_{1,t_{k+1}} & y_{1,t_{k+2}} & \cdots & y_{1,t_{k+N-1}} \\ y_{2,t_k} & y_{2,t_{k+1}} & y_{2,t_{k+2}} & \cdots & y_{2,t_{k+N-1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{H-1,t_k} & y_{H-1,t_{k+1}} & y_{H-1,t_{k+2}} & \cdots & y_{H-1,t_{k+N-1}} \end{bmatrix}_{N\times H} \tag{3.5}$$

In the above equation 3.5, Y_{ref} represents the user-selected segment; t_k and t_{k+N-1} are the start timestamp (at index k) and end timestamp (at index $k+N-1$) of the user-selected segment respectively; Y_{t_k} and $Y_{t_{k+N-1}}$ are the samples at timestamps t_k and t_{k+N-1} respectively; N is number of samples in the user-selected segment.

Let the total number of sliding window segments generated by the VLSW be W and the p^{th} sliding window segment where p ranges from 0 to $(W-1)$ be,

$$Y_p = Y_{t_l,\dots,t_{l+m-1}} = \begin{bmatrix} y_{0,t_l} & y_{0,t_{l+1}} & y_{0,t_{l+2}} & \cdots & y_{0,t_{l+m-1}} \\ y_{1,t_l} & y_{1,t_{l+1}} & y_{1,t_{l+2}} & \cdots & y_{1,t_{l+m-1}} \\ y_{2,t_l} & y_{2,t_{l+1}} & y_{2,t_{l+2}} & \cdots & y_{2,t_{l+m-1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{H-1,t_l} & y_{H-1,t_{l+1}} & y_{H-1,t_{l+2}} & \cdots & y_{H-1,t_{l+m-1}} \end{bmatrix}_{m\times H} \quad (3.6)$$

In the above equation 3.6, Y_p represents the p^{th} sliding window segment where p ranges from 0 to $(W - 1)$ as l ranges from 0 to $(S - 1)$ and m ranges from $(N - d_{min})$ to $(N + d_{max})$ both with the steps of r samples.

For example, if $N = 100,\ r = 5,\ d_{\min} = d_{\max} = 20\%\ of\ N$, then, the sliding window segments will be,

$$Y_{t_0,\dots,t_{79}}, Y_{t_0,\dots,t_{84}}, \dots, Y_{t_0,\dots,t_{114}}, Y_{t_0,\dots,t_{119}}, Y_{t_5,\dots,t_{84}}, Y_{t_5,\dots,t_{89}}, \dots, Y_{t_5,\dots,t_{119}}, Y_{t_5,\dots,t_{124}}, \dots, W\ times.$$

These generated sliding window segments are then compared with the user-selected segment using the customized cost function. For customized cost function, dynamic time warping (DTW) distance is taken here as an example. Let the DTW distance as cost function be,

$$e_p = \mathrm{c}(Y_{ref}, Y_p) = \mathrm{c}(Y_{t_k,\dots,t_{k+N-1}}, Y_{t_l,\dots,t_{l+m-1}}) \quad (3.7)$$

In the above equation 3.7, e_p represents the DTW distance between the user-selected segment Y_{ref} and the p^{th} sliding window segment Y_p where p ranges from 0 to $(W - 1)$ as l ranges from 0 to $(S - 1)$ and m ranges from $(N - d_{min})$ to $(N + d_{max})$ both with the steps of r samples.

For example, if $N = 100,\ r = 5,\ d_{\min} = d_{\max} = 20\%\ of\ N$, then, the DTW distances of each sliding window segments from the user-selected segment will be,

$$e_0 = \mathrm{c}(Y_{t_k,\dots,t_{k+N-1}}, Y_{t_0,\dots,t_{79}}),\ e_1 = \mathrm{c}(Y_{t_k,\dots,t_{k+N-1}}, Y_{t_0,\dots,t_{84}}),\ e_2 = \mathrm{c}(Y_{t_k,\dots,t_{k+N-1}}, Y_{t_0,\dots,t_{89}}), \dots, W\ times.$$

After getting the DTW distances of each sliding window segment from the user-selected segment, a plot with DTW distance of p^{th} sliding window segment from the user-selected segment (e_p) as y-axis and sliding window segment number (p) as x-axis where p ranges from 0 to $(W - 1)$ is generated for visualization to observe the behavior of the DTW distance as a cost function with the VLSW search method. The following Figure 3.14 illustrates this DTW distance of p^{th} sliding window segment from the user-selected segment (e_p) vs sliding window segment number (p) plot. In this plot, the blue-colored trace represents the DTW distance as the cost. Furthermore, a repetitive pattern of lower and higher DTW distances can be observed. Moreover, the number of this repetitive pattern is corresponding to the number of batches because five batches of data are used as well as five repetitive patterns can be seen in this plot. So, it can be concluded that the repetitive pattern of lower DTW distances occurs due to the presence of the sliding window segments that are similar to the user-selected segment in different batches. Similarly, the repetitive pattern of higher DTW distances occurs due to the presence of the sliding window segments that are completely different from the user-selected segment in different batches. But as the objective is to find the sliding window segments similar to the user-selected segment, the main focus is on the repetitive pattern of lower DTW distances.

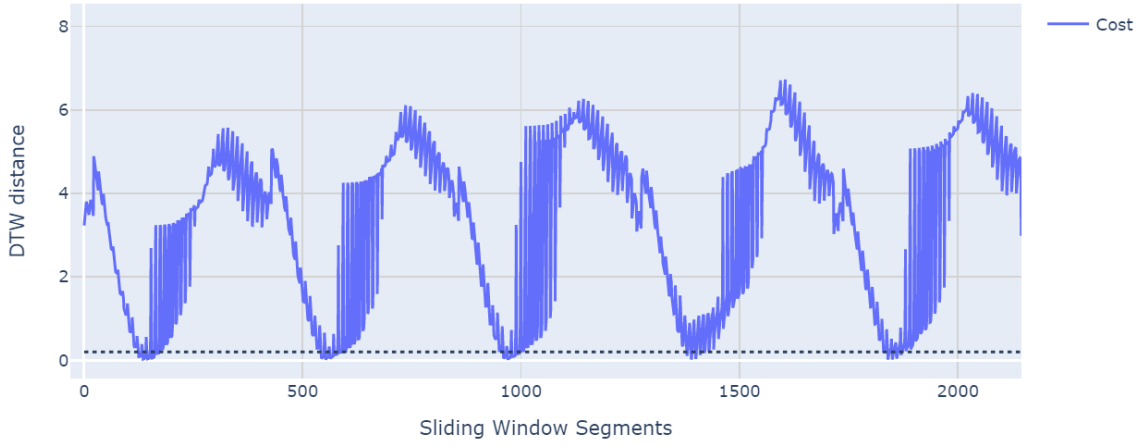

Figure 3.14 DTW distance vs sliding window segments plot.

Another way to visualize the behavior of DTW distances of each sliding window segment from the user-selected segment with the VLSW search method is the contour plot. The following Figure 3.15 illustrates this contour plot where the first index of each sliding window segment is shown on the x-axis, the length of sliding window segments is shown on the y-axis and the DTW distance is shown on the z-axis. The red regions represent the regions of lower DTW distances whereas the blue regions represent the regions of higher DTW distances. In this plot also, a repetitive pattern of regions with lower and higher DTW distances corresponding to the number of batches can be observed. Therefore, this observation also supports that the repetitive pattern of lower DTW distances occurs due to the presence of the sliding window segments that are similar to the user-selected segment in different batches.

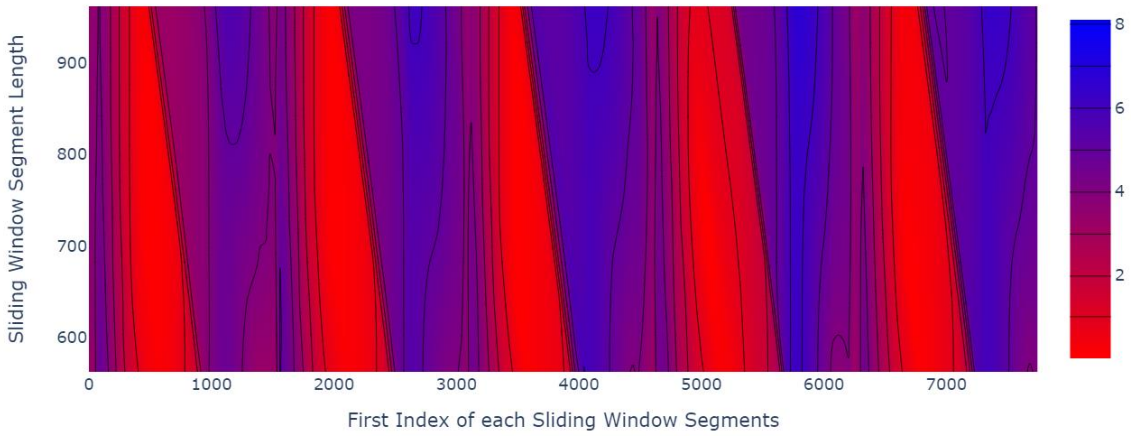

Figure 3.15 DTW distance vs sliding window segments contour plot.

After observing the behavior of the DTW distance as a cost function with the VLSW search method and finding out that the repetitive pattern of lower DTW distances occurs due to the presence of sliding window segments that are similar to the user-selected segment in different batches, the idea is to separate these sliding window segments with lower DTW distances into multiple groups which are far away from one another and then, finding the sliding window segment with lowest DTW distance in each group. In this way, all the sliding window segments similar to the user-selected segment can be detected in different batches.

The sliding window segments with lower DTW distances can be separated into multiple groups which are far away from one another by setting a cost threshold parameter. In this example, the cost threshold parameter is set in such a way that the number of groups generated corresponds to the number of batches. Here, the cost threshold is set to 0.2 to generate five groups for five batches. The following Figure 3.16 illustrates the same DTW distance plot with a cost threshold set to 0.2 represented by the horizontal dashed line. In this figure, the blue-colored trace represents the DTW distance as the cost and the red-colored boxes represent the multiple groups of sliding window segments with lower DTW distances created by setting the cost threshold to 0.2. Moreover, there are five red-colored boxes representing the five groups from G1 to G5.

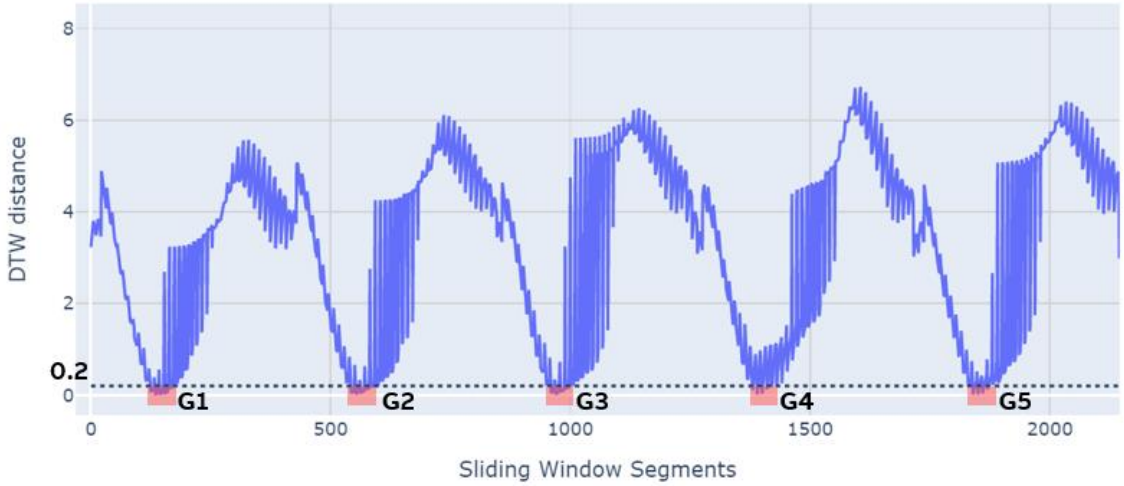

Figure 3.16 Separation of groups by setting the cost threshold.

By setting the cost threshold, the sliding window segments with DTW distance equal to or lower than the cost threshold are selected. Let these selected sliding window segments be the points. Then, the groups of these points which are far from one another are separated using the indices of the points lying at the cost threshold but there are many close points that lie at the cost threshold and indices of not all points lying at the cost threshold can be used to separate the groups otherwise the generated groups will not be far away from one another, and this can lead to the detection of overlapped sliding window segments that are similar to user-selected segment. To overcome this problem, the desired points lying at the cost threshold should be at least some segments away. This can be done by setting a minimum segment distance parameter between the points at the cost threshold before selecting their indices as a candidate for separating the groups. The following Figure 3.17 illustrates the separation of groups using the indices of the desired points lying at the cost threshold. In this figure, the blue-colored trace represents the DTW distance as the cost, the horizontal dashed line represents the cost threshold that is set to 0.2, the red-colored boxes represent the multiple groups of sliding window segments with lower DTW distances created by setting the cost threshold to 0.2, and the green-colored points lying at the cost threshold are the desired points that can be used for separating the groups. The green-colored points are the desired points because they are the start and the end points of each group and are far from one another.

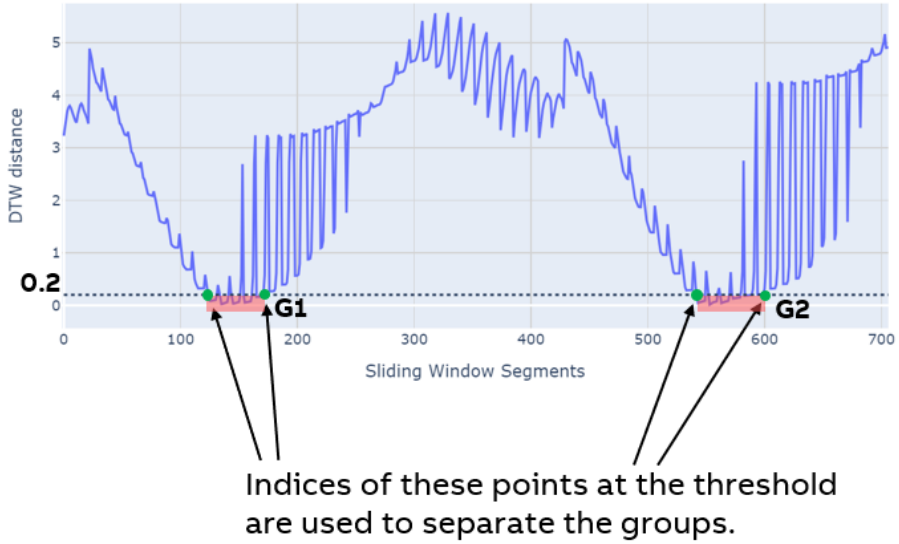

Figure 3.17 Separation of groups using the indices of the desired points lying at the cost threshold.

The following Figure 3.18 illustrates the difference between the desired and the undesired points lying at the cost threshold that can be and cannot be used for separating the groups respectively. In this figure, the blue-colored trace represents the DTW distance as the cost, the horizontal dashed line represents the cost threshold that is set to

0.2, the red-colored box represents the group of sliding window segments with lower DTW distances created by setting the cost threshold to 0.2, the green-colored points lying at the cost threshold are the desired points that can be used for separating the groups and the red-colored point lying at the cost threshold is the undesired point that cannot be used for separating the groups. The green-colored points are the desired points because they are the start and the end points of the group and are far from each other. The red-colored point is undesired because it is close to the start point of the group and indices of not all points lying at the cost threshold can be used to separate the groups otherwise the generated groups will not be far away from one another, and this can lead to the detection of overlapped sliding window segments that are similar to user-selected segment.

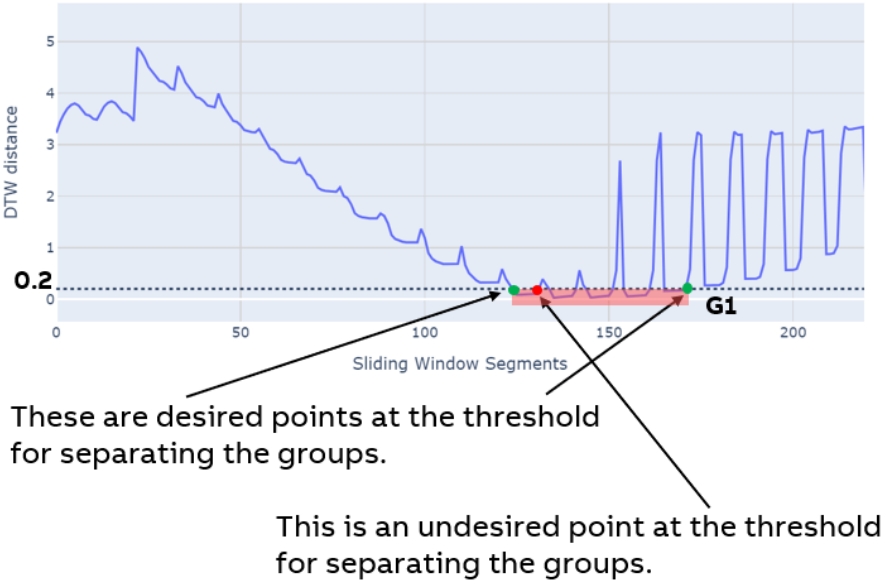

Figure 3.18 Difference between the desired and the undesired points lying at the cost threshold.

After separating the sliding window segments with lower DTW distances into multiple groups which are far away from one another, the sliding window segments with the lowest DTW distances from each group are found. The following Figure 3.19 illustrates the same DTW distance plot with a cost threshold set to 0.2 represented by the horizontal dashed line. In this figure, the blue-colored trace represents the DTW distance as the cost and the red-colored boxes represent the multiple groups of sliding window segments with lower DTW distances created by setting the cost threshold to 0.2.

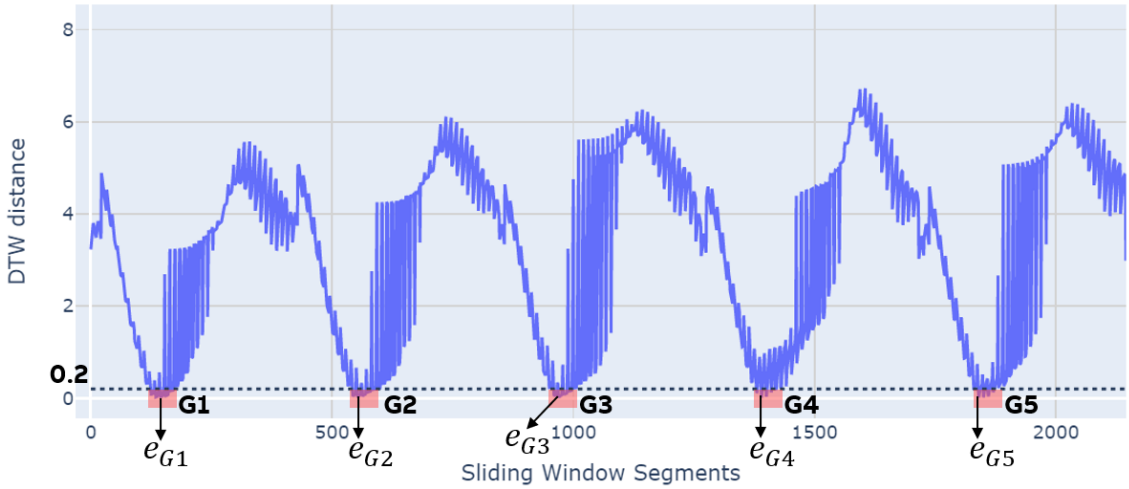

Figure 3.19 Finding the sliding window segments with the lowest DTW distances from each group.

In this example, there are five groups that are generated by setting the cost threshold parameter to 0.2 for five batches. Therefore, there will also be five sliding window segments with the lowest DTW distance from each of these five groups and can be found by using the following equations,

$$e_{G1} = min(e_{G1_1}, e_{G1_2}, e_{G1_3}, \ \dots) \tag{3.8}$$

$$e_{G2} = min(e_{G2_1}, e_{G2_2}, e_{G2_3}, \ \dots) \tag{3.9}$$

$$e_{G3} = min(e_{G3_1}, e_{G3_2}, e_{G3_3}, \ \dots) \tag{3.10}$$

$$e_{G4} = min(e_{G4_1}, e_{G4_2}, e_{G4_3}, \ \dots) \tag{3.11}$$

$$e_{G5} = min(e_{G5_1}, e_{G5_2}, e_{G5_3}, \ \dots) \tag{3.12}$$

In the above equations from 3.8 to 3.12, e_{G1}, e_{G2}, e_{G3}, e_{G4} and e_{G5} represent the five lowest DTW distances from each of these five groups, and by using the indices of these DTW distances, their corresponding five sliding window segments can also be found because information about both the sliding window segments and their corresponding DTW distances are already stored together at the time of iterating through the sliding window segments and calculating their DTW distances from the user-selected segment in the VLSW search method. After finding the sliding window segments similar to the user-selected segment using the lowest DTW distances from each group, the corresponding start and end indices of each of these sliding window segments are returned as the detected change points from the VLSW search method. In this example, ten detected change points corresponding to the five detected sliding window segments will be returned from the VLSW search method.

Throughout the description of the VLSW search method above, five different tuning parameters were introduced apart from the user-selected segment. These five VLSW search method tuning parameters are minimum deviation, maximum deviation, steps, cost threshold, and minimum segment distance. The detailed descriptions of these five tuning parameters are described in the result update sub-section 3.2.8. These tuning parameters are initially set to some default values to obtain the initial results. But after getting the initial results and evaluation, these tuning parameters can again be adjusted by the user to different values in order to get more accurate and updated results.

The algorithm for the VLSW search method is implemented in two steps:

1. As the first step, all the sliding window segments along with their respective DTW distances from the user-selected segment are found using the minimum deviation, the maximum deviation, and the steps tuning parameters.
2. In the second step, the sliding window segments with lower DTW distances are separated into multiple groups which are far away from one another using the cost threshold and minimum segment distance tuning parameters, then, the sliding window segments with the lowest DTW distances from each group are found and finally, the corresponding start and end indices of each of these found sliding window segments are returned as the detected change points.

Moreover, the computation time for the first step is more than that of the second step due to the calculation of DTW distances for a large number of sliding window segments that are generated through the VLSW search method in the first step.

3.2.7 Postprocessing and Evaluation

The detected change points obtained from the interactive CPD approach contain only indices and do not have any timestamps information. Therefore, postprocessing is needed to convert the information from detected change point indices to the detected change point timestamps. Moreover, after getting the detected change points timestamps, the detected segments as well as the artificial labels for the events are also generated through the postprocessing steps. These postprocessing steps are used in the experiments sections 4.1 and 4.2.

The following Figure 3.20 illustrates an example of the result after postprocessing steps obtained from the interactive CPD approach using the VLSW search method and the DTW distance as the cost function. This figure shows the process variable, the ground truth event, the detected change points, and the detected event for multiple batches from a simulated dataset using the interactive CPD with the VLSW search method and the DTW distance as the cost function. The blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. Due to the overlapping of the ground truth events and the detected events, the green

and the orange-colored areas are overlapped and hence, producing the dark green areas. Since these overlapping areas are large, therefore, it can be said that the result in this case is quite good.

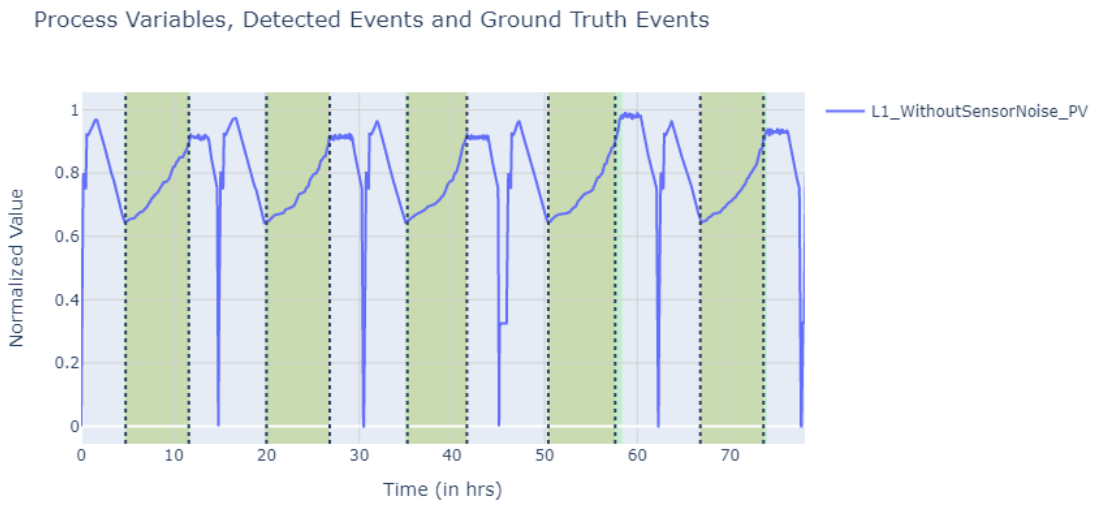

Figure 3.20 Result obtained after the postprocessing steps from the interactive CPD approach using the VLSW search method and the DTW distance as the cost function.

After postprocessing, an initial result is obtained in the form of artificial labels for the events. But to know the accuracy of these results, some evaluation metrics or method are needed. In this thesis, various evaluation metrics and method are implemented depending upon the availability of ground truth events. These evaluation metrics and method are used in the experiments sections 4.1 and 4.2. Moreover, these evaluation metrics and method aim to answer the third research question of this thesis, i.e., how to evaluate and measure the results of interactive CPD.

In case when the ground truth events are present, the following evaluation metrics can be used:

- **Annotation error (AE):**
  Annotation error is defined as the absolute error between the number of true events and the number of detected events. Mathematically,

$$AE = |N_{true} - N_{detected}| \tag{3.13}$$

  In the above equation 3.13, N_{true} represents the number of true events and $N_{detected}$ represents the number of detected events.

- **Mean of true event duration (μ_{true}):**
  Mean of true event duration is defined as the average duration of the ground truth events across all batches. Mathematically,

$$\mu_{true} = \frac{\sum_{i=1}^{N_{true}} D_{i,true}}{N_{true}} \tag{3.14}$$

  In the above equation 3.14, N_{true} represents the number of the ground truth events and $D_{i,true}$ represents the duration of the i^{th} ground truth event.

- **Standard deviation of true event duration (σ_{true}):**
  Standard deviation of true event duration is defined as the standard deviation of the duration of the ground truth events across all batches. Mathematically,

$$\sigma_{true} = \sqrt{\frac{\sum_{i=1}^{Ntrue}(D_{i,true} - \mu_{true})^2}{N_{true}}} \tag{3.15}$$

In the above equation 3.15, N_{true} represents the number of the ground truth events, $D_{i,true}$ represents the duration of the i^{th} ground truth event and μ_{true} represents the mean of true event duration.

- **Mean of detected event duration ($\mu_{detected}$):**
  Mean of detected event duration is defined as the average duration of the detected events across all batches. Mathematically,

$$\mu_{detected} = \frac{\sum_{i=1}^{N_{detected}} D_{i,detected}}{N_{detected}} \tag{3.16}$$

In the above equation 3.16, $N_{detected}$ represents the number of the detected events and $D_{i,detected}$ represents the duration of the i^{th} detected event.

- **Standard deviation of detected event duration ($\sigma_{detected}$):**
  Standard deviation of detected event duration is defined as the standard deviation of the duration of the detected events across all batches. Mathematically,

$$\sigma_{detected} = \sqrt{\frac{\sum_{i=1}^{N_{detected}} (D_{i,detected} - \mu_{detected})^2}{N_{detected}}} \tag{3.17}$$

In the above equation 3.17, $N_{detected}$ represents the number of the detected events, $D_{i,detected}$ represents the duration of the i^{th} detected event and $\mu_{detected}$ represents the mean of detected event duration.

- **Start location error (SLE):**
  Start location error is defined as the average absolute error between the start timestamps of the detected events and the ground truth events. Mathematically,

$$SLE = \frac{\sum_{i=1}^{N_{detected}} \left| t_{i,detected,start} - t_{i,true,start} \right|}{N_{detected}} \tag{3.18}$$

In the above equation 3.18, $N_{detected}$ represents the number of the detected events and $t_{i,detected,start}$ represents the start timestamp of the i^{th} detected event and $t_{i,true,start}$ represents the start timestamp of the ground truth event nearest to the start timestamp of the i^{th} detected event.

- **End location error (ELE):**
  End location error is defined as the average absolute error between the end timestamps of the detected events and the ground truth events. Mathematically,

$$ELE = \frac{\sum_{i=1}^{N_{detected}} \left| t_{i,detected,end} - t_{i,true,end} \right|}{N_{detected}} \tag{3.19}$$

In the above equation 3.19, $N_{detected}$ represents the number of the detected events and $t_{i,detected,end}$ represents the end timestamp of the i^{th} detected event and $t_{i,true,end}$ represents the end timestamp of the ground truth event nearest to the end timestamp of the i^{th} detected event.

- **Duration error (DE):**
  Duration error is defined as the average absolute error between the duration of the detected events and the ground truth events. Mathematically,

$$SLE = \frac{\sum_{i=1}^{N_{detected}} \left| D_{i,detected} - D_{i,true} \right|}{N_{detected}} \tag{3.20}$$

In the above equation 3.20, $N_{detected}$ represents the number of the detected events and $D_{i,detected}$ represents the duration of the i^{th} detected event and $D_{i,true}$ represents the duration of the ground truth event nearest to the i^{th} detected event.

If the ground truth events are absent, then, the evaluation by visualization method can be used.

Evaluation by visualization method: In this evaluation method, all the detected segments are plotted on the same graph together. This visualization helps the user to check how much the detected segments are similar. Using this visualization, the user can then tune the parameters to update the results further and increase the accuracy.

The following Figure 3.21 shows an example of the evaluation by visualization method. In this method, all the detected segments are plotted on the same graph together. The colored traces represent the detected segments. This visualization can be used to evaluate the result if there are no ground truth events are present.

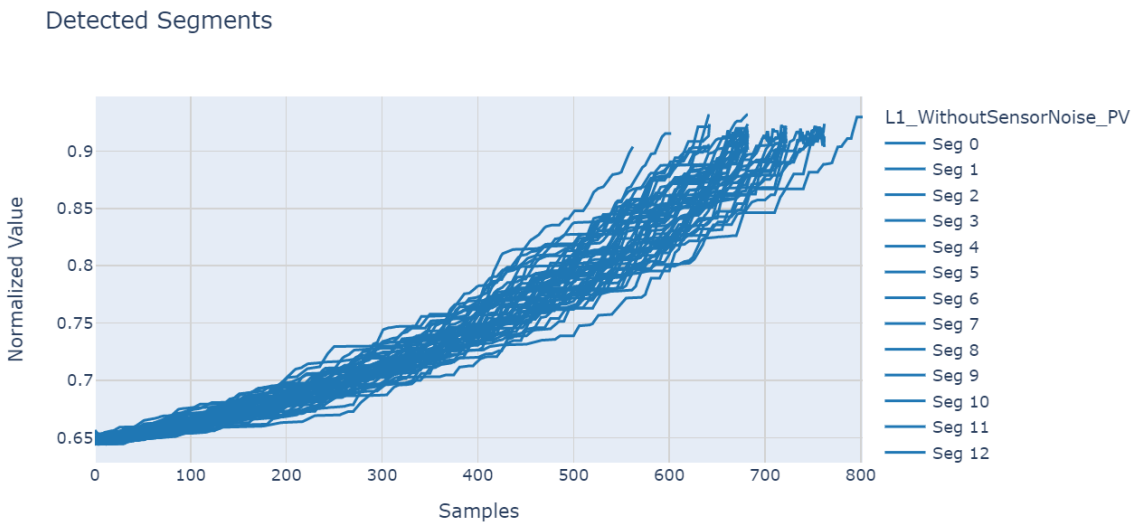

Figure 3.21 An example of the evaluation by visualization method.

3.2.8 Result Update

After getting the initial results from the interactive CPD approach, an evaluation of the results is performed. With the help of evaluation, parameter tuning is done through user feedback to further update and increase the accuracy of the results as well as get the updated artificial labels for the events.

In the case of the PELT search method, penalty value can be used as a tuning parameter to further update and increase the accuracy of the results because the number of change points decreases with an increase in penalty value and vice versa [1] [3].

In the case of the VLSW search method, five different tuning parameters can be used to further update and increase the accuracy of the results. These five tuning parameters are following:

- **Minimum deviation (d_{min}):**
  From equation 3.1, it can be said that increasing the minimum deviation will decrease the minimum length of the sliding window segments, covering cases with smaller segment lengths and can increase the accuracy of the result. But according to the explanation of equation 3.6, it will also increase the number of sliding window segments for comparison and computation speed.

- **Maximum deviation (d_{max}):**
  From equation 3.2, it can be said that increasing the maximum deviation will increase the maximum length of the sliding window segments, covering cases with larger segment lengths and can increase the accuracy of the result. But according to the explanation of equation 3.6, it will also increase the number of sliding window segments for comparison and computation speed.

- **Steps (r):**

From the explanation of equation 3.6, it can be said that decreasing the steps size can increase the accuracy of the results because more appropriate sliding window segments will be generated but it will also increase the number of sliding window segments for comparison and computation speed.

- **Cost threshold:**
  From Figure 3.16, it can be said that changing the cost threshold will instantly change the number of detected change points. If the cost threshold is very low, then, some groups will be filtered out and a small number of change points will be obtained. If the cost threshold is very high, then, most of the sliding window segments will lie in a few larger groups and a small number of change points will be obtained. But if the cost threshold is in the middle, then, there will be more groups and a large number of change points will be obtained.

- **Minimum segment distance:**
  From Figure 3.17 and Figure 3.18, it can be said that increasing the minimum segment length will instantly increase the distance between detected segments and can avoid the problem of overlapped detected segments.

Moreover, these five tuning parameters of the VLSW search method take different times to update the results. This means that the changes in the minimum deviation, the maximum deviation, and the steps parameter will take some time to update the results as both steps of the algorithm for the VLSW search method will run again for these changes. But changes in the minimum segment distance and the cost threshold parameters will update the results instantly because only the second step of the algorithm for the VLSW search method will run for these changes. A detailed explanation of both these steps can be found in search methods sub-section 3.2.6.

The interactive change point detection approach is based on user feedback and hence, a user interface (UI) prototype is needed for the users to visualize and incorporate their feedback into the algorithm. In the next section 3.3, the implementation of this UI prototype is discussed.

3.3 Prototype Implementation

A user interface (UI) prototype is developed in this thesis to implement the complete interactive process involved in the interactive change point detection approach. The interactive CPD approach is based on user interaction and feedback. This includes providing suitable visualizations to the user as well as incorporating user feedback into the algorithm for getting both the initial and the updated results. Therefore, there is a need for a user interface (UI) prototype where the users can have suitable visualizations and also can give their valuable feedback to obtain both the initial and the updated results. By developing the prototype, the second research question of this thesis was answered, i.e., how to provide the best visualization and incorporate user feedback into interactive CPD to obtain both initial and updated results. There are two parts of this prototype: front-end and back-end.

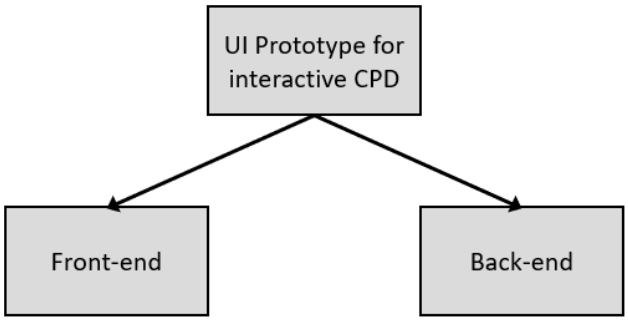

Figure 3.22 Two parts of UI prototype for interactive change point detection.

As shown in Figure 3.22, the UI prototype for interactive CPD is developed in front-end and back-end parts. The functionalities for the data visualizations and the user interactions are implemented in the front-end part and the implementations of the preprocessing steps, the interactive CPD algorithm, the postprocessing steps, and the

evaluation are performed in the back-end part. The front-end part communicates to the back-end part through the application programming interface (API) requests and responses. In this section, detailed descriptions of the development of both the front-end and the back-end parts are discussed.

3.3.1 Front-end

In the UI prototype, the front-end is the user-end or client-side which is shown to the users and also used to interact with them. All the visualizations, user input fields, and other user interaction functionalities are implemented in the front-end part of the prototype. To develop front-end for interactive CPD, the following programming language, frameworks, and libraries are used:

- javaScript
- Node.js and npm
- Vue.js
- Vuex
- Vuetify
- axios
- plotly.js

The following four functionalities are implemented at the front-end part of the prototype:

1. Data management.
2. Data visualization and event annotation.
3. Evaluation and result update.
4. Saving the results and exporting the artificial labels for the events.

Data management
In the prototype, the users should be able to load their time series data files containing different process variables and events data. Therefore, the data management tab under the front-end application provides the facility for the users to upload their data files and also to select a particular data file for further operations. The following Figure 3.23 shows the screenshot of this data management tab. In this figure, the user input field for uploading the data file and the user selection tab for selecting the data file can be seen. The data file uploaded here as an example contains the simulated dataset [34].

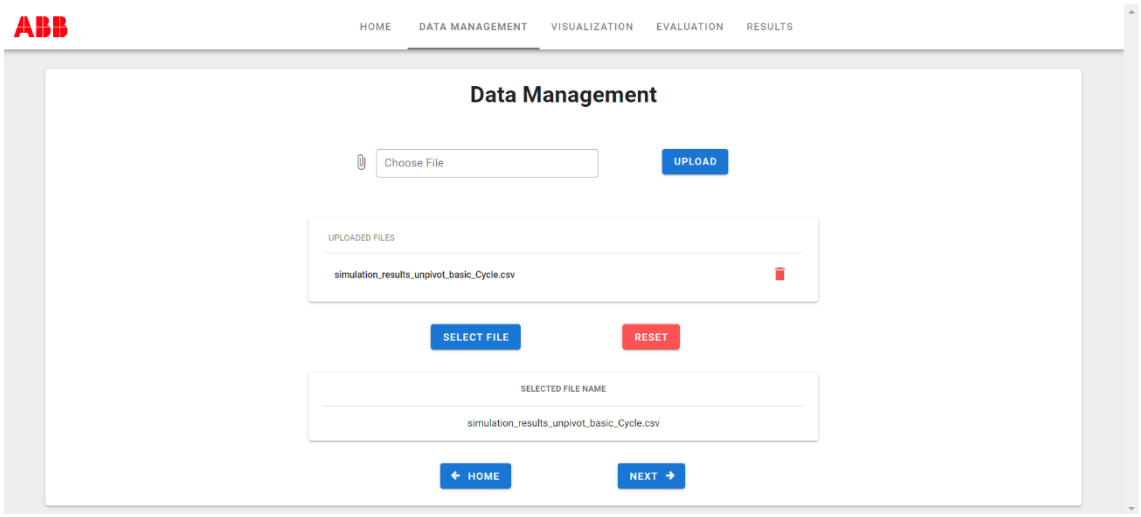

Figure 3.23 Screenshot of the data management tab of the prototype.

Data visualization and event annotation
The users should be able to visualize the time series data containing the process variables and the events. Therefore, the visualization tab under the front-end application provides the facility for the users to select multiple process variables and the event they want to visualize and after their selection, the plot is generated. This visualization also helps the users to decide the process variables for generating artificial labels for a particular event. The following Figure 3.24 shows the screenshot of this visualization tab for selecting the process variables, the event, and the

range of data to be visualized. In this figure, the user input fields for selecting the process variables, the event, and the range of data to be visualized can be seen.

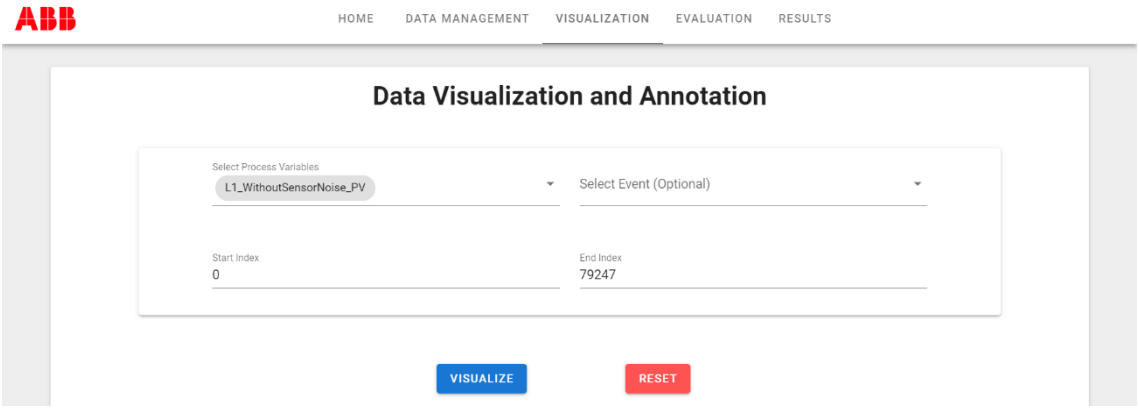

Figure 3.24 Screenshot of the visualization tab for selecting the process variables, the event, and the range of data to be visualized.

The following Figure 3.25 shows the screenshot of the visualization tab for visualizing the process variables and the event. In this figure, the blue-colored trace represents the selected process variable from the simulated dataset. In this example, the ground truth event is not selected and visualized here in order to show that the interactive CPD algorithm can also work without the presence of the ground truth events.

Figure 3.25 Screenshot of the visualization tab for visualizing the process variables and the event.

Moreover, there should be some functionality for the users to select a segment because incorporating user feedback in the form of the user-selected segment into the interactive change point detection algorithm is a very important step. Therefore, under this same visualization tab, the facility for the users to annotate and generate artificial labels for a particular event is provided. In this tab, the users can select a segment from the same visualization plot as well as they can select the process variables that will be used for artificial events generation. Additionally, they can give an event name for the generated artificial labels. After generating the initial artificial labels, an initial result can also be seen in the same visualization plot. The following Figure 3.26 shows the screenshot of this visualization tab for annotating the event through the user-selected segment. In this figure, the blue-colored trace represents the process variable and the light blue-colored area with the blue border represents the user-selected segment. The user input field for selecting the process variables that will be used for creating the artificial labels for the event as well as the field for giving the event name as the label can be seen. In the prototype, the users can select a segment of their interest on the visualization plot with the help of a mouse.

Figure 3.26 Screenshot of the visualization tab for annotating the event through the user-selected segment.

The following Figure 3.27 shows the screenshot of the visualization tab for visualizing the initial result. In this figure, the blue-colored trace represents the process variable, the vertical dotted lines represent the detected change points and orange-colored areas represent the detected events.

Figure 3.27 Screenshot of the visualization tab for visualizing the initial result.

Evaluation and result update

After getting the initial result, the users should be able to evaluate this result and then, should be able to update them also. Therefore, the evaluation tab under the front-end application provides the facility for the users to evaluate the result through evaluation by visualization method. This evaluation method is used in the prototype because there was a need for an evaluation method that will work irrespective of the presence of the ground truth events and this evaluation by visualization method can be used to evaluate the result even if there are no ground truth events are present. More details of this evaluation method can be found in the postprocessing and evaluation subsection 3.2.7. The following Figure 3.28 shows the screenshot of this evaluation tab that shows the evaluation by visualization method. In this figure, the colored traces represent the detected segments.

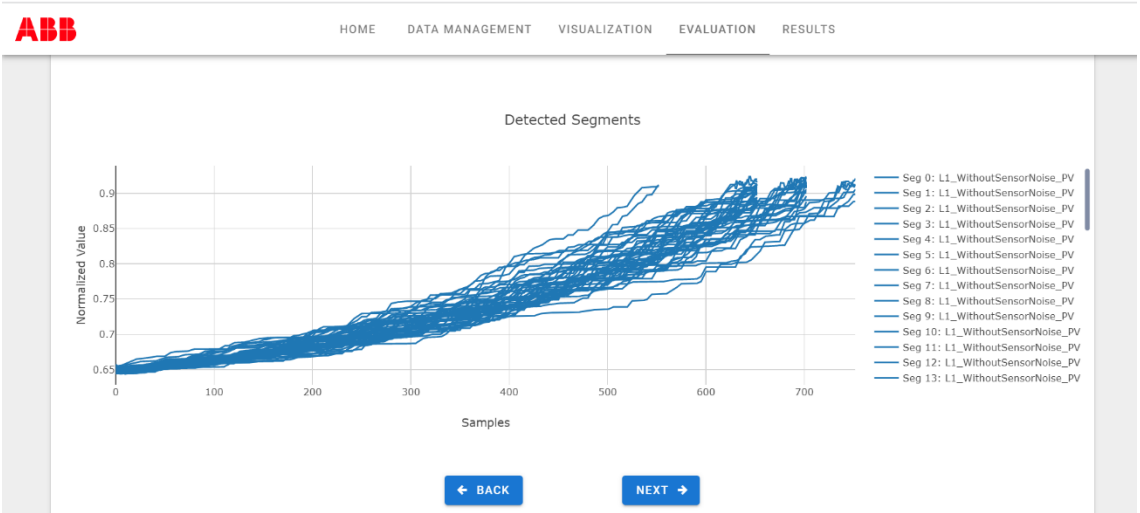

Figure 3.28 Screenshot of the evaluation tab that shows the evaluation by visualization method.

In the same tab, after evaluation, there is a facility for the users to adjust the five tuning parameters of the VLSW search method using sliders to further update the results. The changes in minimum deviation, the maximum deviation, and the steps parameter will take time to update the results as the complete algorithm will run again for these changes. But changes in the minimum distance and the cost threshold parameters will reflect instantly in the results because only a part of the algorithm will run for these changes. More details about all these five tuning parameters of the VLSW search method can be found in the result update sub-section 3.2.8. The following Figure 3.29 shows the screenshot of the evaluation tab for tuning the different parameters of the VLSW search method to update the results. In this figure, the minimum deviation, the maximum deviation, and the steps parameters are shown which will take time to update the results.

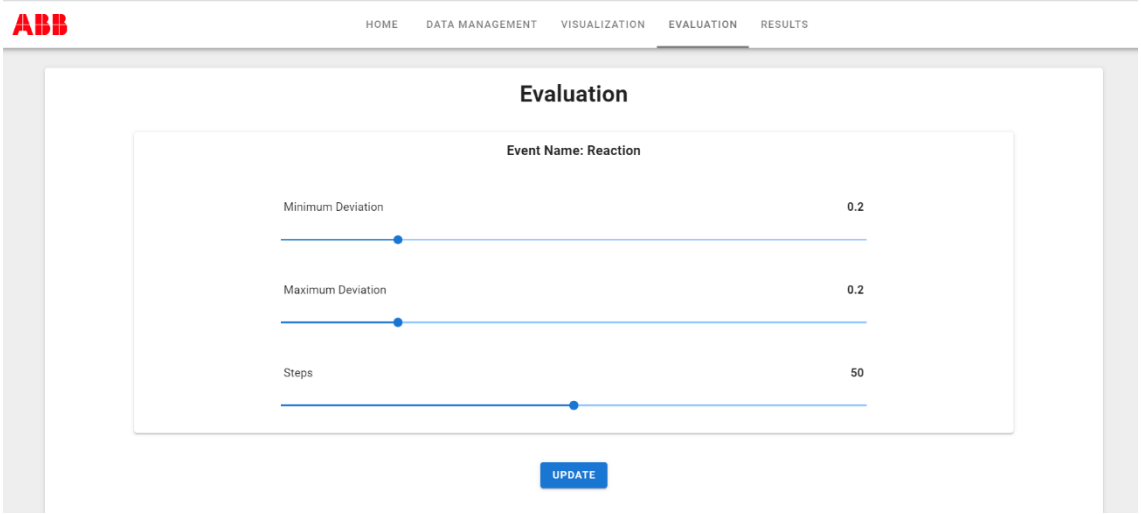

Figure 3.29 Screenshot of the evaluation tab for tuning the different VLSW search method parameters.

The following Figure 3.30 shows the screenshot of the evaluation tab for tuning the different parameters of the VLSW search method to update the results. In this figure, minimum distance and the cost threshold parameters are shown which will instantly update the results just by changing the values through the sliders.

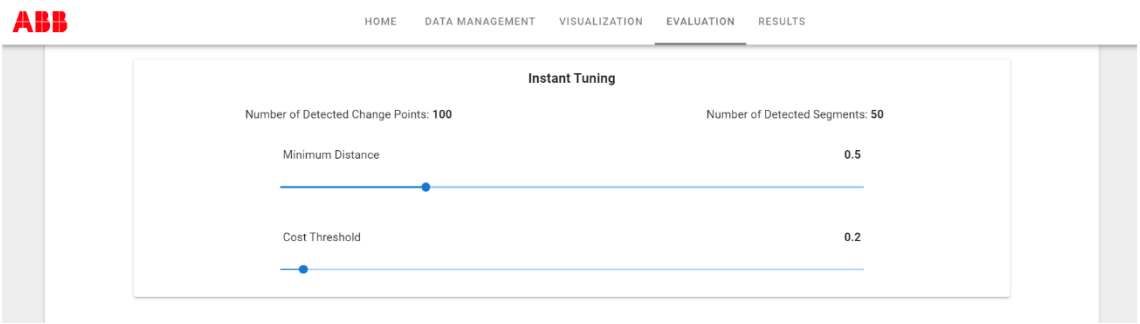

Figure 3.30 Screenshot of the evaluation tab for instantly tuning the different VLSW search method parameters.

Saving the results and exporting the artificial labels for the events

After the users are satisfied with the result, they should be able to save their results, delete the saved results and export the generated artificial labels for the events. Therefore, the results tab under the front-end application provides the facility for the users to save their results, delete the saved results and also export the generated artificial labels for the events as a comma-separated value (CSV) file. The following Figure 3.31 shows the screenshot of this result tab for saving the results. In this figure, the blue-colored trace represents the process variable, the vertical dotted lines represent the detected change points and orange-colored areas represent the detected events.

Figure 3.31 Screenshot of the result tab for saving the results.

The following Figure 3.32 shows the screenshot of the result tab for exporting the generated artificial labels for the events. In this figure, a table with the details of a saved result, the facility to select the saved results for delete/export, and the exported result in a CSV file can be seen.

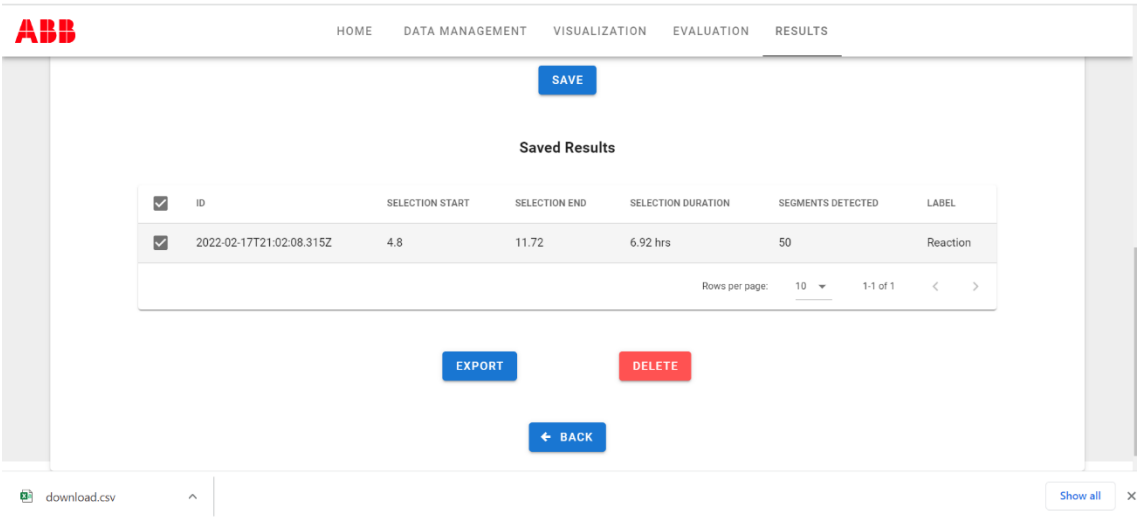

Figure 3.32 Screenshot of the result tab for exporting the generated artificial labels for the events.

3.3.2 Back-end

In the UI prototype, the back-end is the server-side where all the application programming interface (API) requests from the front-end are handled and the corresponding responses are forwarded back to the front-end. All the preprocessing steps, implementation of the interactive CPD algorithm, postprocessing steps, and evaluation are implemented in the back-end part of the prototype. To develop back-end for interactive CPD, the following programming language, frameworks, and libraries are used:

- Python
- pip
- FastAPI
- Uvicorn
- pydantic
- NumPy
- pandas
- scikit-learn
- tslearn
- Poetry

The following four functionalities are implemented at the back-end part of the prototype:

1. **Preprocessing steps:**
   The preprocessing steps like down-sampling and normalization are performed in the back-end part of the UI prototype. More details of the preprocessing steps can be found in sub-section 3.2.3.

2. **Interactive CPD algorithm:**
   The interactive CPD algorithm using the VLSW search method and the DTW distance as the cost function is implemented in the back-end part of the UI prototype. More details of the interactive CPD algorithm can be found in section 3.2.

3. **Postprocessing steps:**
   The postprocessing steps like converting the detected change point indices into detected change point timestamps as well as generating artificial labels for the events are performed in the back-end part of the UI prototype. More details of the preprocessing steps can be found in sub-section 3.2.7.

4. **Evaluation:**
   The evaluation by visualization method is implemented in the back-end part of the UI prototype. More details of this evaluation method can be found in sub-section 3.2.7.

4 Result

In this chapter, the experiments with interactive change point detection for the events identification on different datasets along with their results are discussed. There are two datasets that are used for the experiments: the simulated dataset and the real-life dataset. There are two sections in this chapter: the experiments, the evaluations, and the results of the interactive change point detection with the simulated dataset and the real-life dataset are respectively discussed in the first and the second section.

4.1 Experiments on Simulated Dataset

In this section, the simulated dataset is used for the experiments with interactive change point detection. This simulated dataset is generated by developing and simulating a benchmark model of a batch process with full control over disturbances and noises. The details of this simulated dataset can be found in [34].

In the experiments with interactive change point detection on the simulated dataset, two preprocessing steps are used: down-sampling and normalization. Down-sampling is used to speed up the computation and normalization is used to transform the values of all the process variables on the same scale. The following figures Figure 4.1 and Figure 4.2 illustrate the preprocessed multiple process variables and the events from both single batch and multiple batches respectively of the simulated dataset. The colored traces represent the process variables and colored areas represent the multiple events in each batch. In Figure 4.2, five batches of data are shown.

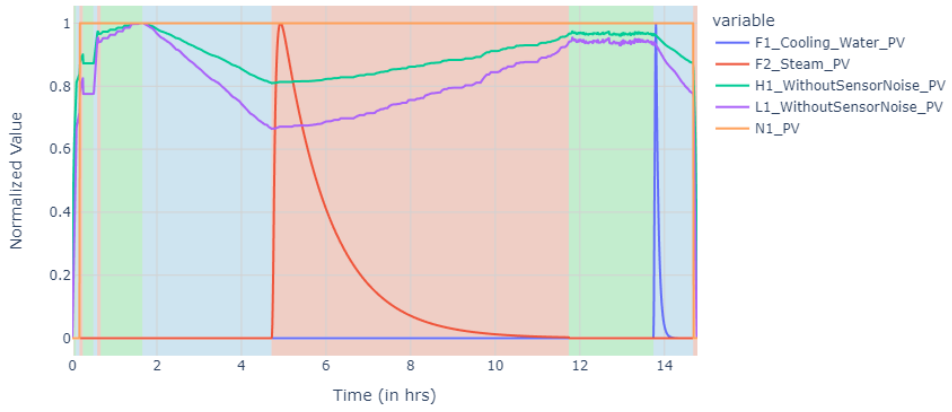

Figure 4.1 Multiple process variables and events from a single batch of the simulated dataset.

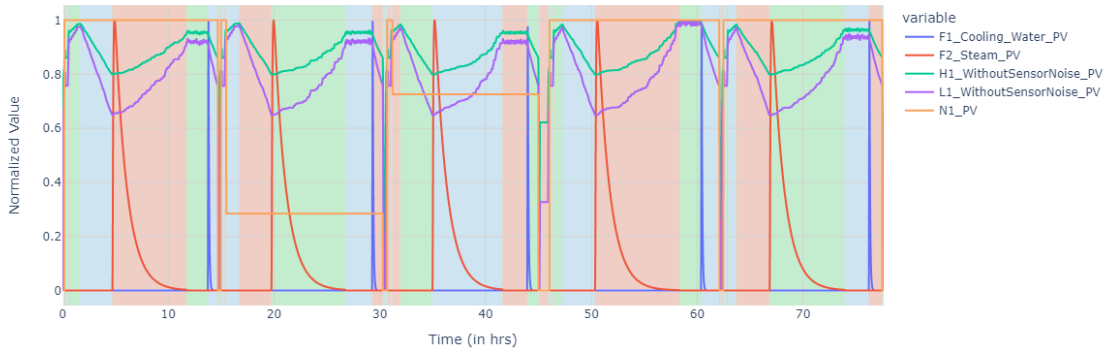

Figure 4.2 Multiple process variables and events from multiple batches of the simulated dataset.

4.1.1 Experiments Setup

In the experiments with interactive change point detection for events identification on the simulated dataset, different combinations of the search methods, the customized cost functions, and the tuning parameters are used. For search method, linearly penalized segmentation using pruned exact linear time (PELT) and varying length sliding window (VLSW) are used; for customized cost function, prediction error and dynamic time warping (DTW) distance are used; and finally, for tuning parameters, a constant value penalty function for PELT and VLSW search method parameters are used. More details of these methods and functions can be found in section 3.2.

For events identification with interactive change point detection, two sets of experiments are performed here. The following Table 4.1 describes these two sets of experiments. In this table, the columns represent the name of the experiment sets, the description of the experiment sets, the dataset with the number of batches used as well as whether the univariate/multivariate data is used for the experiment sets, the events for which the experiments in the experiment sets are performed, the search methods used, the customized cost functions used, the tuning parameters used, and the evaluation metrics/methods used respectively.

Table 4.1 Description of two different sets of experiments performed with interactive change point detection for events identification.

S. No.	Experiment Set	Description	Dataset	Events	Search method	Customized cost functions	Tuning parameters	Evaluation
1.	Experiment S1	To get the interactive CPD results for multiple batch data with the PELT search method and different customized cost functions by manually optimizing (fixed) penalty value to get close to the optimum or actual number of change points.	Simulated Dataset (50 batches of data; univariate)	Reaction	PELT	• Prediction error from a linear model • Prediction error from a non-linear model • DTW distance	Constant penalty function	• Annotation error • Mean of true event duration • Standard deviation of true event duration • Mean of detected event duration • Standard deviation of detected event duration • Start location error • End location error • Duration error • Evaluation by visualization
2.	Experiment S2	To get the interactive CPD results for multiple batch data with the VLSW search method and the DTW distance as the customized cost function by tuning different parameters using user feedback to get artificial labels for the events accurately.	Simulated Dataset (50 batches of data; both univariate and multivariate)	• Product Transfer • Reaction • Post Reaction • Add Educt 1	VLSW	DTW distance	VLSW search method parameters	• Annotation error • Mean of true event duration • Standard deviation of true event duration • Mean of detected event duration • Standard deviation of detected event duration • Start location error • End location error • Duration error • Evaluation by visualization

Each of these two sets of experiments contains several experiments composed by taking different combinations of the customized cost functions, the events, and whether the data is univariate or multivariate. The detailed descriptions of all these experiments are following:

Experiment S1
The aim of this first set of experiments is to get the interactive CPD results for multiple batch data with the PELT search method and different customized cost functions by manually optimizing (fixed) penalty value to get close to the optimum or actual number of change points. The dataset used for these experiments is the simulated dataset containing 50 batches of data. Moreover, univariate data is used in different experiments of this set. For the search method, PELT is used; for customized cost functions, prediction error from a linear model, prediction error from a non-linear model, and DTW distance is used; for tuning parameters, the constant penalty function is used. More details on this search method and the customized cost functions can be found in sub-sections 3.2.6 and 3.2.5 respectively. Along with these, different evaluation metrics/method are also used for evaluating the results. These evaluation metrics and method are Annotation error, Mean of true event duration, Standard deviation of true event duration, Mean of detected event duration, Standard deviation of detected event duration, Start location error, End location error, Duration error, and Evaluation by visualization method. All detailed descriptions of these evaluation metrics and method can be found in sub-section 3.2.7.

For all the experiments in this first set, the following steps are followed:

- First, the data file of the simulated dataset is read. In this example, this data file contains both the process variables data and the events data.
- Then, these time series data from the data file are preprocessed. The sampling time of the time series data is 0.001 hour. It is down-sampled to 0.01 hour to speed up the computation.
- After down-sampling, the range of the data or the number of batches is selected for the experiment.
- Then, the event and the process variable(s) are selected for the experiment and the selected process variable(s) is/are normalized.
- In all the experiments, the ground truth events are used for selecting the user-selected segment as well as for obtaining the evaluation metrics. After normalization, the first ground truth event is taken as the user-selected segment.
- Then, using this user-selected segment, a linear or a non-linear model is trained based on the characteristics of the process variable contained in the user-selected segment. The linear regression and the polynomial regression are used here for the linear and the non-linear model respectively. Moreover, for calculation of the cost, either the prediction error from the trained model is used or the DTW distance from the user-selected segment is used.
- Then, the initial or default value to the tuning parameter of the PELT search method, i.e., the penalty value is assigned.
- After initializing the value for the tuning parameter, the assigned penalty value, the complete dataset along with either the trained model or the user-selected segment depending upon the used customized cost function are passed into the algorithm for interactive CPD with the PELT search method to get the initial detected change point indices.
- Then, these detected change point indices are converted to the detected change point timestamps and then, to the detected events through the postprocessing steps. The conversion of the detected change point timestamps to the detected events also requires either the trained model or the user-selected segment depending upon the used customized cost function. The costs for all the events formed through these change points are then calculated using the used customized cost function and half of these events with the lowest cost are selected for the detected events.
- Finally, different evaluation metrics are calculated based on the ground truth events. Apart from these metrics, the evaluation by visualization method is also used for evaluating the result but this method does not require ground truth events. More details on these evaluation metrics and method can be found in sub-section 3.2.7. The results can be updated after getting the evaluation by tuning the parameter to increase the accuracy. The PELT search method has some more parameters which can be tuned along with the penalty value. The minimum size parameter is also used in the experiments along with the penalty value to increase the accuracy of the results.

The number of experiments performed under this first set of experiments is 3. The following Table 4.2 describes these 3 experiments. In this table, the columns represent the name of the experiments, the dataset with the number of batches used as well as whether the univariate/multivariate data is used for the experiments, the event for which the experiments are performed, the process variable(s) used for the experiments (single process variable in the

univariate case and multiple process variables in the multivariate case), the search method used, the customized cost function used, the values from the used tuning parameters respectively.

Table 4.2 Description of experiments under the first set of experiments.

S. No.	Experiment	Dataset	Event	Used process variable(s)	Search method	Customized cost function	Tuning parameters
1.	Experiment S1.1	Simulated Dataset (50 batches of data; univariate)	Reaction	L1_WithoutSensorNoise_PV	PELT	Prediction error from a linear model (linear regression model)	Penalty value = 3; Minimum size = 600
2.	Experiment S1.2	Simulated Dataset (50 batches of data; univariate)	Reaction	L1_WithoutSensorNoise_PV	PELT	Prediction error from a non-linear model (polynomial regression model of degree 2)	Penalty value = 3; Minimum size = 600
3.	Experiment S1.3	Simulated Dataset (50 batches of data; univariate)	Reaction	L1_WithoutSensorNoise_PV	PELT	DTW distance	Penalty value = 0.1; Minimum size = 600

Experiment S2

The aim of this second set of experiments is to get the interactive CPD results for multiple batch data with the VLSW search method and the DTW distance as the customized cost function by tuning different parameters using user feedback to get artificial labels for the events accurately. The dataset used for these experiments is the simulated dataset containing 50 batches of data. Moreover, both univariate and multivariate data are used in different experiments of this set. For the search method, VLSW is used; for customized cost function, DTW distance is used; for tuning parameters, VLSW search method parameters are used which are minimum deviation, maximum deviation, steps, cost threshold, and minimum segment distance. More details on this search method, the customized cost function, and the tuning parameters can be found in sub-sections 3.2.6, 3.2.5, and 3.2.8 respectively. Along with these, different evaluation metrics/method are also used for evaluating the results. These evaluation metrics and method are Annotation error, Mean of true event duration, Standard deviation of true event duration, Mean of detected event duration, Standard deviation of detected event duration, Start location error, End location error, Duration error, and Evaluation by visualization method. All detailed descriptions of these evaluation metrics and method can be found in sub-section 3.2.7.

For all the experiments in this second set, the following steps are followed:

- First, the data file of the simulated dataset is read. In this example, this data file contains both the process variables data and the events data.
- Then, these time series data from the data file are preprocessed. The sampling time of the time series data is 0.001 hour. It is down-sampled to 0.01 hour to speed up the computation.
- After down-sampling, the range of the data or the number of batches is selected for the experiment.
- Then, the event and the process variable(s) are selected for the experiment and the selected process variable(s) is/are normalized.
- In all the experiments, the ground truth events are used for selecting the user-selected segment as well as for obtaining the evaluation metrics. After normalization, the first ground truth event is taken as the user-selected segment.
- Then, the initial or default values to all the tuning parameters of the VLSW search method which are minimum deviation, maximum deviation, steps, cost threshold, and minimum segment distance are assigned. More details on these tuning parameters can be found in sub-section 3.2.8. Moreover, for setting the value of the minimum segment distance parameter, a different parameter is introduced, i.e., the minimum distance parameter. This is done because the variation in the value range of this minimum segment

distance parameter is very large due to its dependency on other parameters such as the length of the user-selected segment, the minimum deviation, the maximum deviation, and the steps. Therefore, it is difficult to know the initial value range of the minimum segment distance parameter, and hence, a mathematical formula is proposed for it,

$$Minimum\ segment\ distance = \frac{Minimum\ distance\ \times (N_{max} - N_{min}) \times N_{user}}{r^2} \quad (4.1)$$

In the above equation 4.1, N_{max} represents the maximum length of the sliding window, N_{min} represents the minimum length of the sliding window, N_{user} represents the length of the user-selected segment and r represents the steps size. Therefore, by setting the small value range for the minimum distance parameter (e.g., from 0 to 1), the minimum segment distance parameter can be easily initialized.

- After initializing the values for the tuning parameters, these parameters along with the user-selected segment and complete dataset are passed into the algorithm for interactive CPD with VLSW search method and DTW distance as the cost function to get the initial detected change point indices.
- Then, these detected change point indices are converted to the detected change point timestamps and then, to the detected events through the postprocessing steps. The conversion of change point timestamps into the detected events is easy in the case of the VLSW search method as the detected change points were originally obtained from similar detected segments. Therefore, the number of detected change points will always be even, and by making a pair of every two change points (e.g., first and second change point, third and fourth change point, fifth and sixth change point, and so on), the detected events can be obtained. The details of this search method can be found in sub-section 3.2.6.
- Finally, different evaluation metrics are calculated based on the ground truth events. Apart from these metrics, the evaluation by visualization method is also used for evaluating the result but this method does not require ground truth events. More details on these evaluation metrics and method can be found in sub-section 3.2.7. The results can be updated after getting the evaluation by tuning the parameters to increase the accuracy.

The number of experiments performed under this second set of experiments is 8. The following Table 4.3 describes these 8 experiments. In this table, the columns represent the name of the experiments, the dataset with the number of batches used as well as whether the univariate/multivariate data is used for the experiments, the event for which the experiments are performed, the process variable(s) used for the experiments (single process variable in the univariate case and multiple process variables in the multivariate case), the search method used, the customized cost function used, the values from the used tuning parameters respectively.

Table 4.3 Description of experiments under the second set of experiments.

S. No.	Experiment	Dataset	Event	Used process variable(s)	Search method	Customized cost function	Tuning parameters
1.	Experiment S2.1	Simulated Dataset (50 batches of data; univariate)	Product Transfer	L1_WithoutSensorNoise_PV	VLSW	DTW distance	• Minimum deviation = 0.2 • Maximum deviation = 0.2 • Steps = 10 • Cost threshold = 0.2 • Minimum distance = 0.5; Minimum segment distance = 186
2.	Experiment S2.2	Simulated Dataset (50 batches of data; multivariate)	Product Transfer	• F1_Cooling_Water_PV • F2_Steam_PV • H1_WithoutSensorNoise_PV • L1_WithoutSensorNoise_PV	VLSW	DTW distance	• Minimum deviation = 0.2 • Maximum deviation = 0.2 • Steps = 10 • Cost threshold = 0.2 • Minimum distance = 0.5; Minimum segment distance = 186

3.	Experiment S2.3	Simulated Dataset (50 batches of data; univariate)	Reaction	L1_WithoutSensorNoise_PV	VLSW	DTW distance	• Minimum deviation = 0.2 • Maximum deviation = 0.4 • Steps = 40 • Cost threshold = 0.2 • Minimum distance = 0.5; Minimum segment distance = 92
4.	Experiment S2.4	Simulated Dataset (50 batches of data; multivariate)	Reaction	• F1_Cooling_Water_PV • F2_Steam_PV • H1_WithoutSensorNoise_PV • L1_WithoutSensorNoise_PV	VLSW	DTW distance	• Minimum deviation = 0.2 • Maximum deviation = 0.2 • Steps = 10 • Cost threshold = 1 • Minimum distance = 0.5; Minimum segment distance = 186
5.	Experiment S2.5	Simulated Dataset (50 batches of data; univariate)	Post Reaction	L1_WithoutSensorNoise_PV	VLSW	DTW distance	• Minimum deviation = 0.2 • Maximum deviation = 0.6 • Steps = 10 • Cost threshold = 0.2 • Minimum distance = 0.5; Minimum segment distance = 160
6.	Experiment S2.6	Simulated Dataset (50 batches of data; multivariate)	Post Reaction	• F1_Cooling_Water_PV • F2_Steam_PV • H1_WithoutSensorNoise_PV • L1_WithoutSensorNoise_PV	VLSW	DTW distance	• Minimum deviation = 0.2 • Maximum deviation = 0.6 • Steps = 10 • Cost threshold = 0.2 • Minimum distance = 0.5; Minimum segment distance = 160
7.	Experiment S2.7	Simulated Dataset (50 batches of data; univariate)	Add Educt 1	L1_WithoutSensorNoise_PV	VLSW	DTW distance	• Minimum deviation = 0.2 • Maximum deviation = 0.6 • Steps = 1 • Cost threshold = 0.2 • Minimum distance = 0.5; Minimum segment distance = 10
8.	Experiment S2.8	Simulated Dataset (50 batches of data; multivariate)	Add Educt 1	• F1_Cooling_Water_PV • F2_Steam_PV • H1_WithoutSensorNoise_PV • L1_WithoutSensorNoise_PV	VLSW	DTW distance	• Minimum deviation = 0.2 • Maximum deviation = 0.6 • Steps = 1 • Cost threshold = 0.2 • Minimum distance = 0.5; Minimum segment distance = 10

4.1.2 Results

There are two sets of experiments that are performed with interactive change point detection for the identification of the events on the simulated dataset. The descriptions of these two sets of experiments are described in the experiments setup sub-section 4.1.1 and the results are presented in this sub-section.

Results of Experiment S1
There are 3 experiments in the first set of experiments. The results of all these 3 experiments are presented below:

- **Experiment S1.1**
  The following Figure 4.3 shows the process variable and the ground truth event from the first batch. In this figure, the blue-colored trace represents the process variable, and the green-colored area represents the ground truth event. This first ground truth event is also taken as the user-selected segment for this experiment.

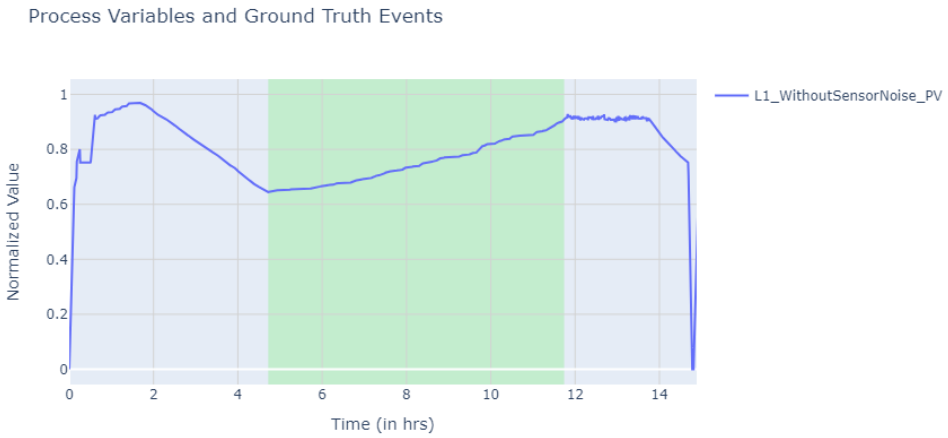

Figure 4.3 Process variable and the ground truth event from the first batch in Experiment S1.1.

The following Figure 4.4 shows the process variable, the ground truth event, the detected change points, and the detected event for all 50 batches. In this figure, the blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. From this figure, it can be observed that the events are detected for all 50 batches. Since the result is not clear here due to a large number of batches, a zoomed-in version of this figure for the first 5 batches is shown in the next Figure 4.5.

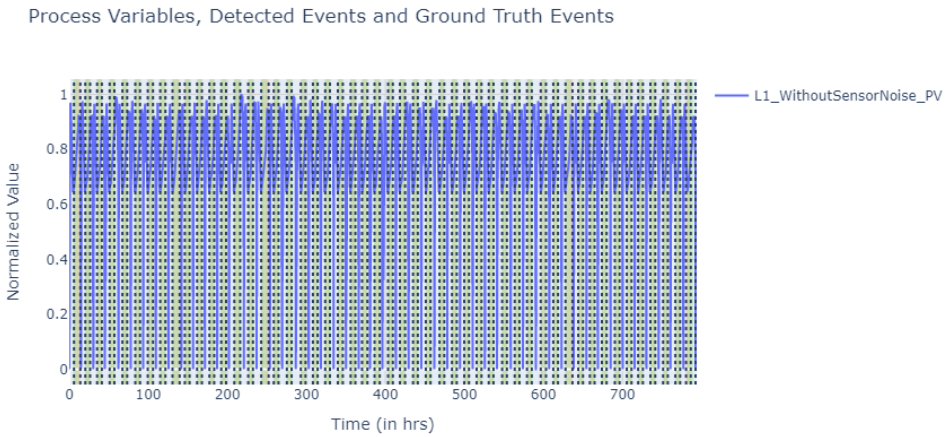

Figure 4.4 Process variable, the ground truth event, the detected change points, and the detected event for all 50 batches in Experiment S1.1.

The following Figure 4.5 shows the process variable, the ground truth event, the detected change points, and the detected event for the first 5 batches. The blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. Due to the overlapping of the ground truth events and the detected events, the green and the orange-colored areas are overlapped and hence, producing the dark green areas. Since these overlapping areas are large, therefore, it can be said that the result in this case is good.

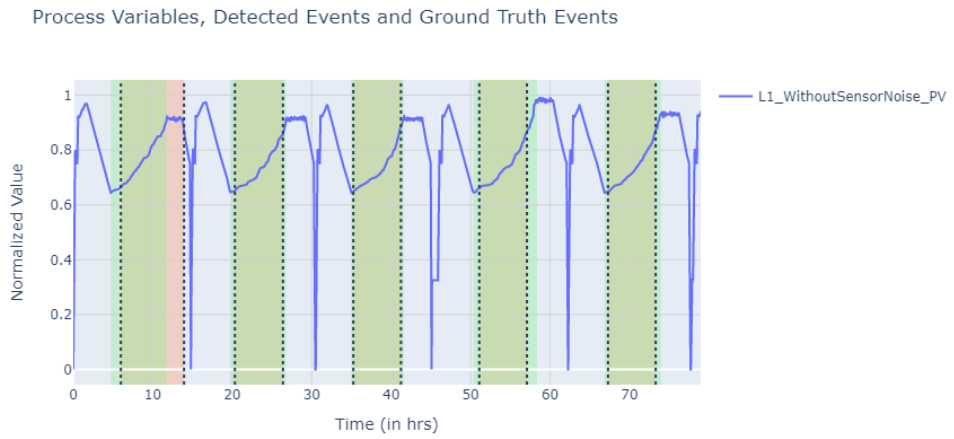

Figure 4.5 Process variable, the ground truth event, the detected change points, and the detected event for the first 5 batches in Experiment S1.1.

The following Figure 4.6 shows the evaluation by visualization method for this experiment where all the detected segments are plotted on the same graph together. In this figure, the colored traces represent all the detected segments. From this figure, it can be observed that most of the detected segments are similar except the detected segments from some batches, especially from the first and the last batch. To improve the result from the first and the last batch, zero padding at the start and at the end of the dataset can be applied. In this experiment, there are 50 ground truth events in 50 batches. Also, 50 detected events and 101 detected change points are obtained from the experiment. So, overall, it can be said that the result from this experiment is good.

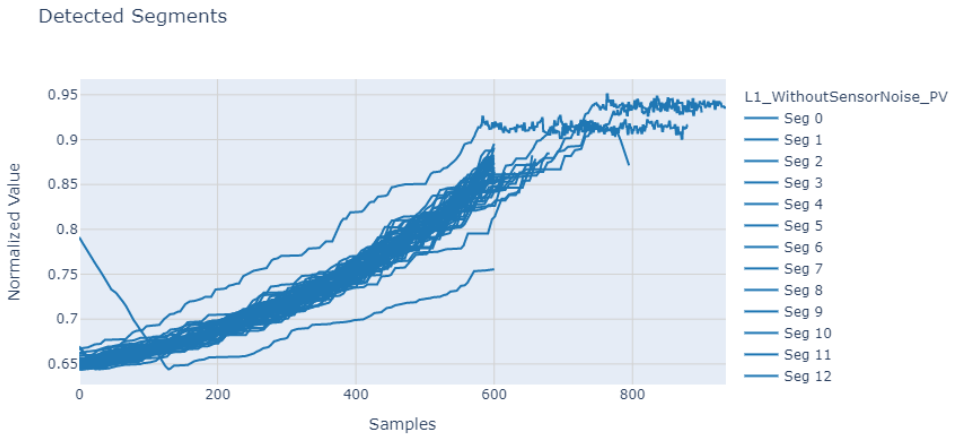

Figure 4.6 Evaluation by visualization method in Experiment S1.1.

- **Experiment S1.2**
  The following Figure 4.7 shows the process variable and the ground truth event from the first batch. In this figure, the blue-colored trace represents the process variable, and the green-colored area represents

the ground truth event. This first ground truth event is also taken as the user-selected segment for this experiment.

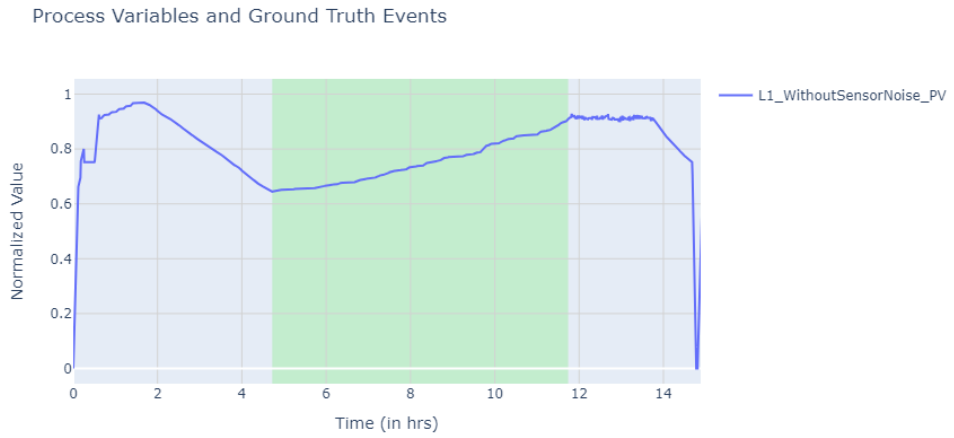

Figure 4.7 Process variable and the ground truth event from the first batch in Experiment S1.2.

The following Figure 4.8 shows the process variable, the ground truth event, the detected change points, and the detected event for all 50 batches. In this figure, the blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. From this figure, it can be observed that the events are detected for all 50 batches. Since the result is not clear here due to the large number of batches, a zoomed-in version of this figure for first 5 batches is shown in the next Figure 4.9.

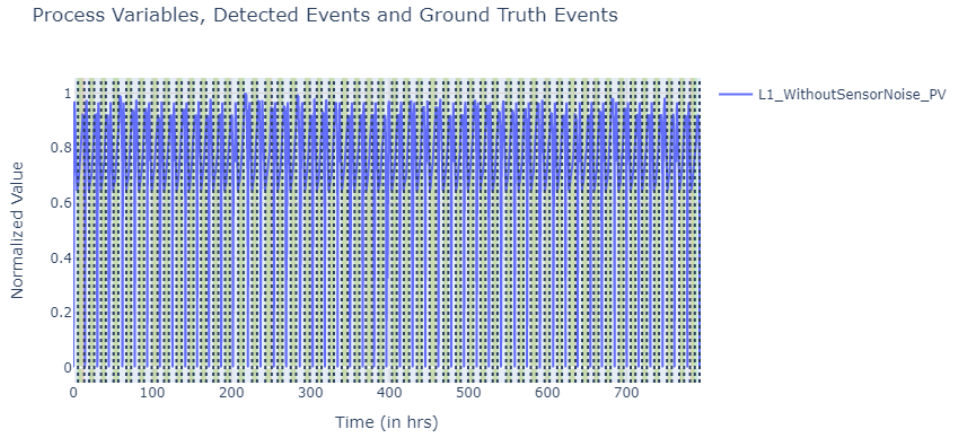

Figure 4.8 Process variable, the ground truth event, the detected change points, and the detected event for all 50 batches in Experiment S1.2.

The following Figure 4.9 shows the process variable, the ground truth event, the detected change points, and the detected event for first 5 batches. The blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. Due to the overlapping of the ground truth events and the detected events, the green and the orange-colored areas are overlapped and hence, producing the dark green areas. Since these overlapping areas are large, therefore, it can be said that the result in this case is good.

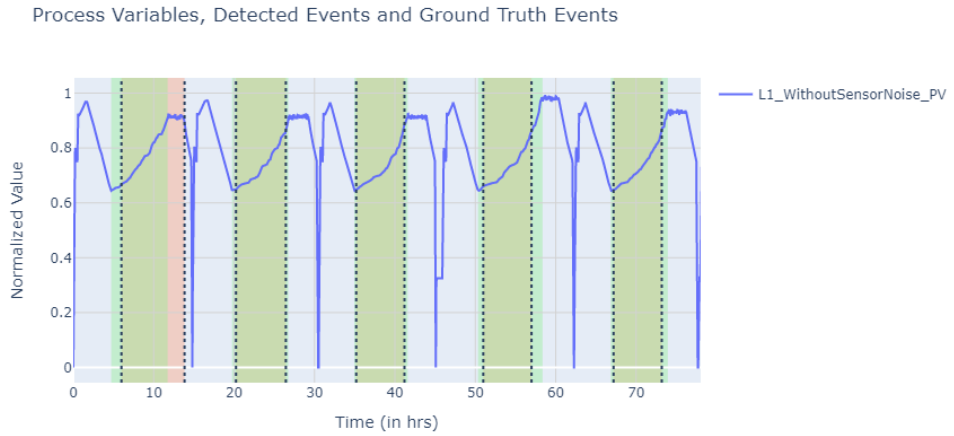

Figure 4.9 Process variable, the ground truth event, the detected change points, and the detected event for first 5 batches in Experiment S1.2.

The following Figure 4.10 shows the evaluation by visualization method for this experiment where all the detected segments are plotted on the same graph together. In this figure, the colored traces represent all the detected segments. From this figure, it can be observed that most of the detected segments are similar except the detected segments from some batches, especially from the first and the last batch. To improve the result from the first and the last batch, zero padding at the start and at the end of the dataset can be applied. In this experiment, there are 50 ground truth events in 50 batches. Also, 50 detected events and 101 detected change points are obtained from the experiment. So, overall, it can be said that the result from this experiment is good.

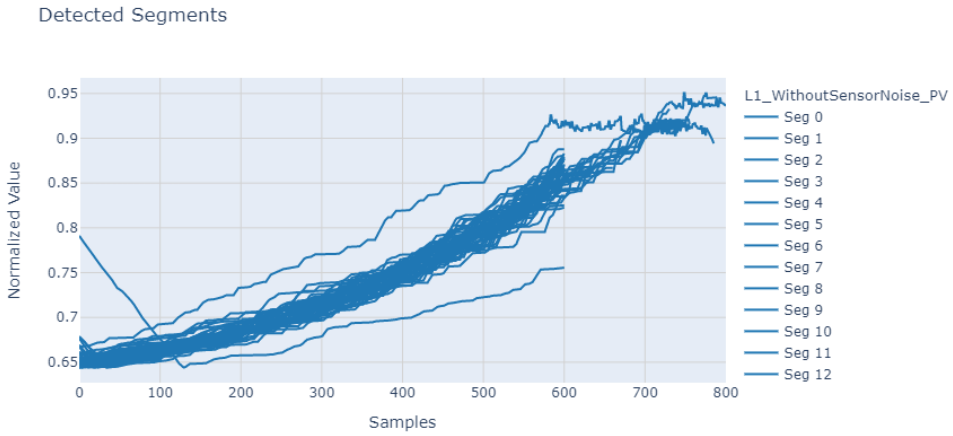

Figure 4.10 Evaluation by visualization method in Experiment S1.2.

- **Experiment S1.3**
  The following Figure 4.11 shows the process variable and the ground truth event from the first batch. In this figure, the blue-colored trace represents the process variable, and the green-colored area represents the ground truth event. This first ground truth event is also taken as the user-selected segment for this experiment.

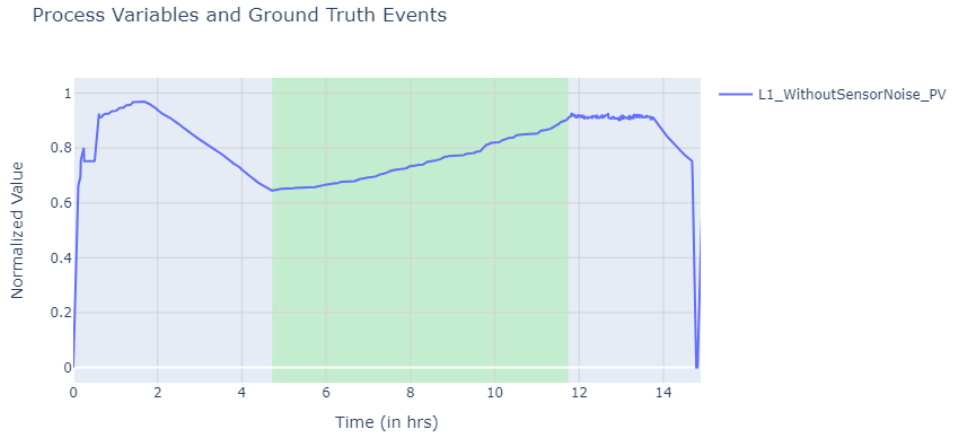

Figure 4.11 Process variable and the ground truth event from the first batch in Experiment S1.3.

The following Figure 4.12 shows the process variable, the ground truth event, the detected change points, and the detected event for all 50 batches. In this figure, the blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. From this figure, it can be observed that the events are detected for all 50 batches. Since the result is not clear here due to the large number of batches, a zoomed-in version of this figure for first 5 batches is shown in the next Figure 4.13.

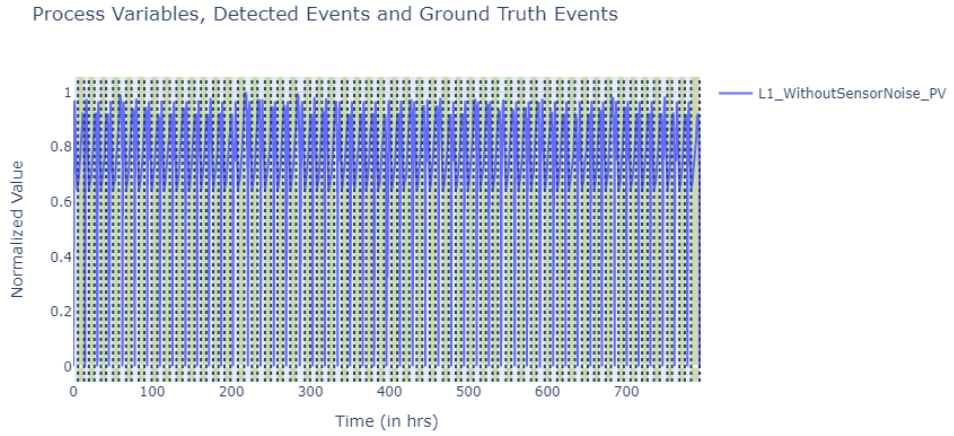

Figure 4.12 Process variable, the ground truth event, the detected change points, and the detected event for all 50 batches in Experiment S1.3.

The following Figure 4.13 shows the process variable, the ground truth event, the detected change points, and the detected event for first 5 batches. The blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. Due to the overlapping of the ground truth events and the detected events, the green and the orange-colored areas are overlapped and hence, producing the dark green areas. Since these overlapping areas are large, therefore, it can be said that the result in this case is good.

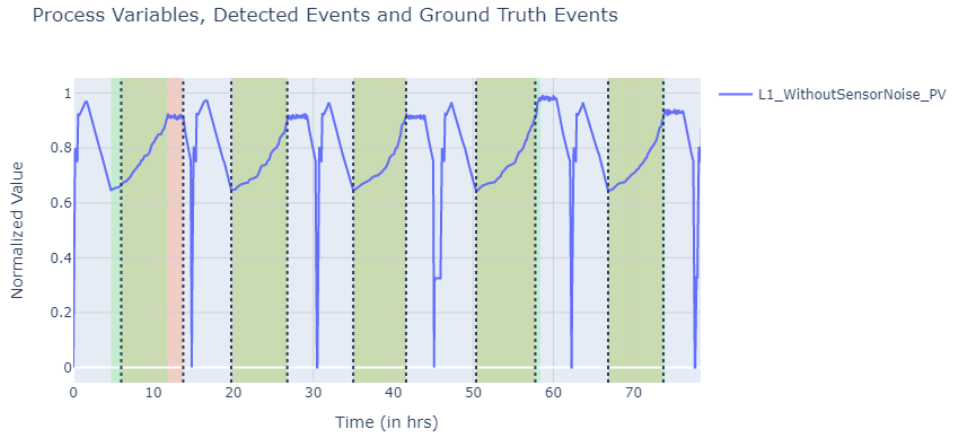

Figure 4.13 Process variable, the ground truth event, the detected change points, and the detected event for first 5 batches in Experiment S1.3.

The following Figure 4.14 shows the evaluation by visualization method for this experiment where all the detected segments are plotted on the same graph together. In this figure, the colored traces represent all the detected segments. From this figure, it can be observed that most of the detected segments are very similar except the detected segments from the first and the last batch. To improve the result from the first and the last batch, zero padding at the start and at the end of the dataset can be applied. In this experiment, there are 50 ground truth events in 50 batches. Also, 50 detected events and 100 detected change points are obtained from the experiment. So, overall, it can be said that the result from this experiment is good.

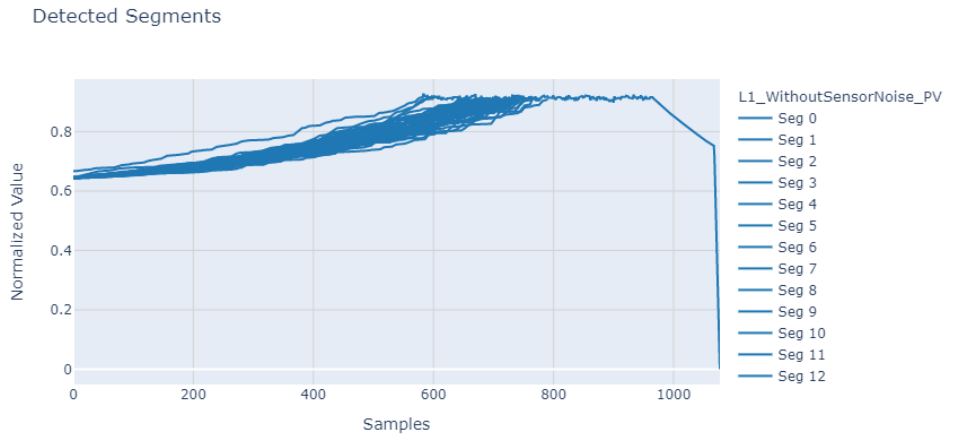

Figure 4.14 Evaluation by visualization method in Experiment S1.3.

The following Table 4.4 demonstrates the summary of results from the first set of experiments. In this table, the columns represent the name of the experiments, the event for which the experiments are performed, the experiments with the univariate/multivariate case, the values from the used evaluation metrics, and the time taken to compute the results respectively. Moreover, the percentage in start location error, end location error, and duration error are calculated over the mean of true event duration.

Table 4.4 Summary of results from the first set of experiments.

Experiment	Event Name	Univariate / Multivariate	True Event Duration Information		Detected Event Duration Information		Annotation Error = | N(true events) – N(detected events) |	Start location error (between true and detected events)		End location error (between true and detected events)		Duration error (between true and detected events)		Computation Time
			Mean [hrs]	Std [hrs]	Mean [hrs]	Std [hrs]		[hrs]	[%]	[hrs]	[%]	[hrs]	[%]	
Experiment S1.1	**Reaction**	**Univariate**	7.2574	0.4598	6.345	0.7817	0	0.3986	5.49	0.9046	12.46	1.1092	15.28	2m 3.9s
Experiment S1.2	**Reaction**	**Univariate**	7.2574	0.4598	6.394	0.6036	0	0.3538	4.88	0.8472	11.67	0.973	13.41	5m 40.7s
Experiment S1.3	**Reaction**	**Univariate**	7.2574	0.4598	7.1074	0.6479	0	0.04	0.55	0.3408	4.69	0.3144	4.33	11m 47.1s

The results from the first set of experiments show that the interactive CPD algorithm with the PELT search method and the DTW distance as the cost function for events identification has achieved good performance for the event Reaction as compared to when the prediction errors from the trained linear and non-linear model are taken as the cost functions. There is no annotation error, indicating that the number of detected events matches exactly with the number of events. The duration error and location errors are also relatively small. On the other hand, the performance for the event Reaction when the prediction errors from the trained linear and non-linear model are taken as the cost functions is not as good as the performance when the DTW distance is taken as the cost function. Although there is no annotation error, the duration error and location errors are relatively large for both cases. The reason for this is the DTW distance considers the shape and the amplitude of the process variable for comparison irrespective of the length of the segment but the prediction error from the trained models can give more cost if the length of the segment under comparison is too large or too small. Moreover, for the event Reaction, by comparing the performance of two prediction errors, it can be said that the non-linear model works slightly better than the linear model. The reason for this is the characteristics of the process variable in the event Reaction is non-linear. Furthermore, in all these experiments with the PELT search method, the results from the first and the last batch are not accurate. However, to improve the result from the first and the last batch, zero padding at the start and at the end of the dataset can be applied. Additionally, the computation time was more for the cost function with the DTW distance than that of the prediction error.

Results of Experiment S2

There are 8 experiments in the second set of experiments. The results of all these 8 experiments are presented below:

- **Experiment S2.1**

  The following Figure 4.15 shows the process variable and the ground truth event from the first batch. In this figure, the blue-colored trace represents the process variable, and the green-colored area represents the ground truth event. This first ground truth event is also taken as the user-selected segment for this experiment.

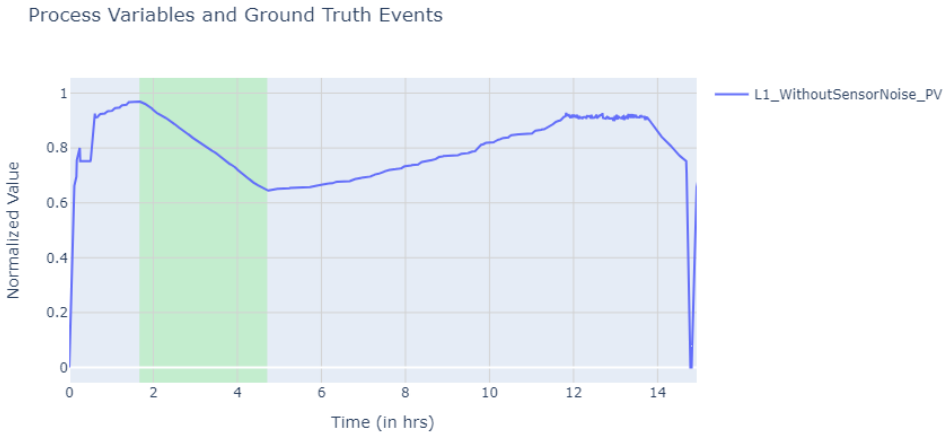

Figure 4.15 Process variable and the ground truth event from the first batch in Experiment S2.1.

The following Figure 4.16 shows the process variable, the ground truth event, the detected change points, and the detected event for all 50 batches. In this figure, the blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the

ground truth events and the orange-colored areas represent the detected events. From this figure, it can be observed that the events are detected for all 50 batches. Since the result is not clear here due to the large number of batches, a zoomed-in version of this figure for first 5 batches is shown in the next Figure 4.17.

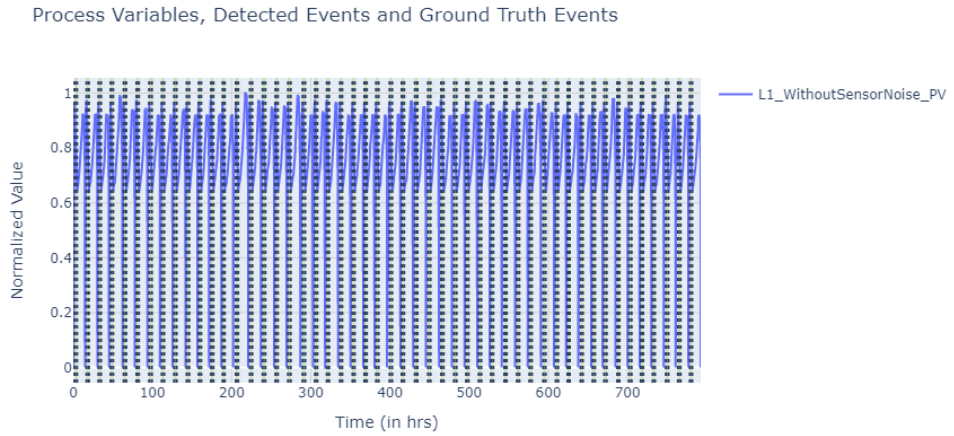

Figure 4.16 Process variable, the ground truth event, the detected change points, and the detected event for all 50 batches in Experiment S2.1.

The following Figure 4.17 shows the process variable, the ground truth event, the detected change points, and the detected event for first 5 batches. The blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. Due to the overlapping of the ground truth events and the detected events, the green and the orange-colored areas are overlapped and hence, producing the dark green areas. Since these overlapping areas are large, therefore, it can be said that the result in this case is quite good.

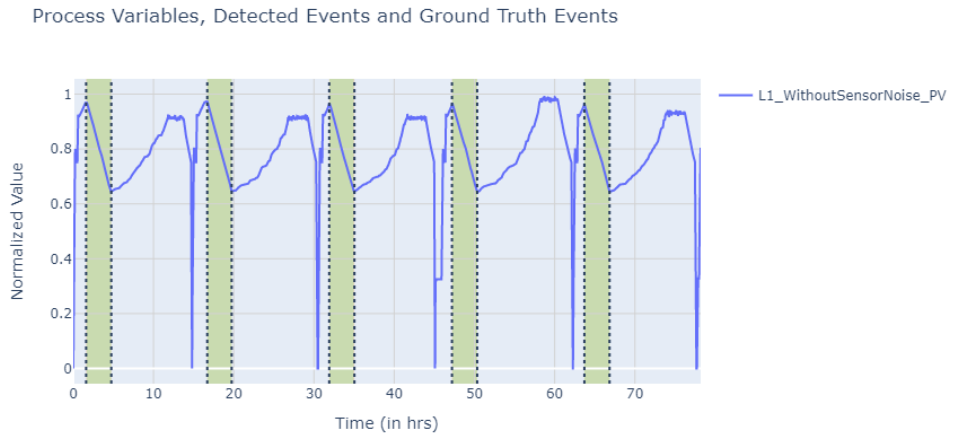

Figure 4.17 Process variable, the ground truth event, the detected change points, and the detected event for first 5 batches in Experiment S2.1.

The following Figure 4.18 shows the evaluation by visualization method for this experiment where all the detected segments are plotted on the same graph together. In this figure, the colored traces represent all the detected segments. From this figure, it can be observed that all the detected segments are very similar. In this experiment, there are 50 ground truth events in 50 batches. Also, 50 detected events and 100 detected change points are obtained from the experiment. So, overall, it can be said that the result from this experiment is quite good.

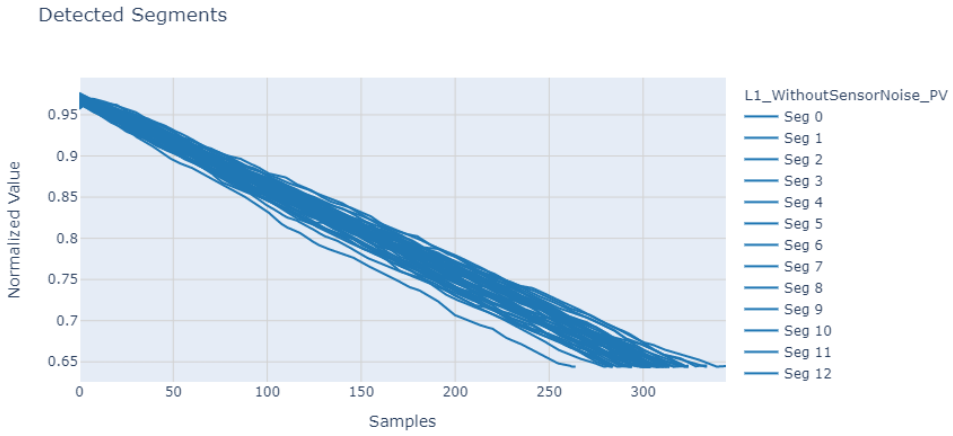

Figure 4.18 Evaluation by visualization method in Experiment S2.1.

- **Experiment S2.2**
  The following Figure 4.19 shows the multiple process variables and the ground truth event from the first batch. In this figure, the colored traces represent the process variables, and the green-colored area represents the ground truth event. This first ground truth event is also taken as the user-selected segment for this experiment.

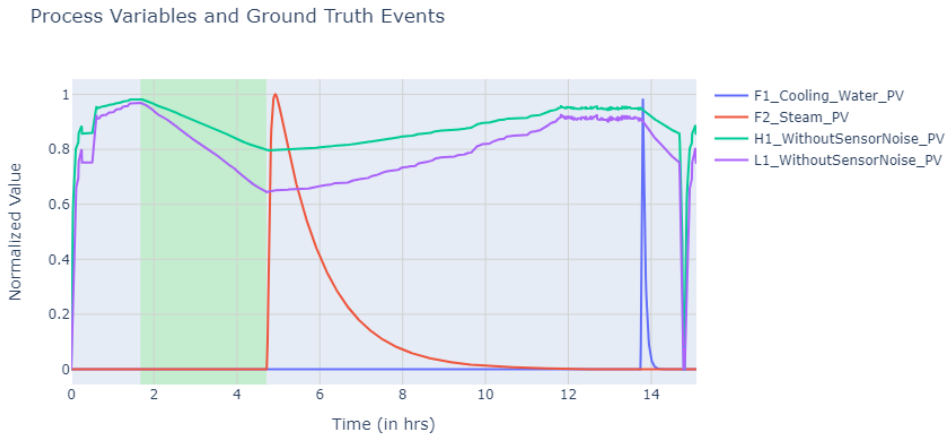

Figure 4.19 Multiple process variables and the ground truth event from the first batch in Experiment S2.2.

The following Figure 4.20 shows the multiple process variables, the ground truth event, the detected change points, and the detected event for all 50 batches. In this figure, the colored traces represent the process variables, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. From this figure, it can be observed that the events are detected for all 50 batches. Since the result is not clear here due to the large number of batches, a zoomed-in version of this figure for first 5 batches is shown in the next Figure 4.21.

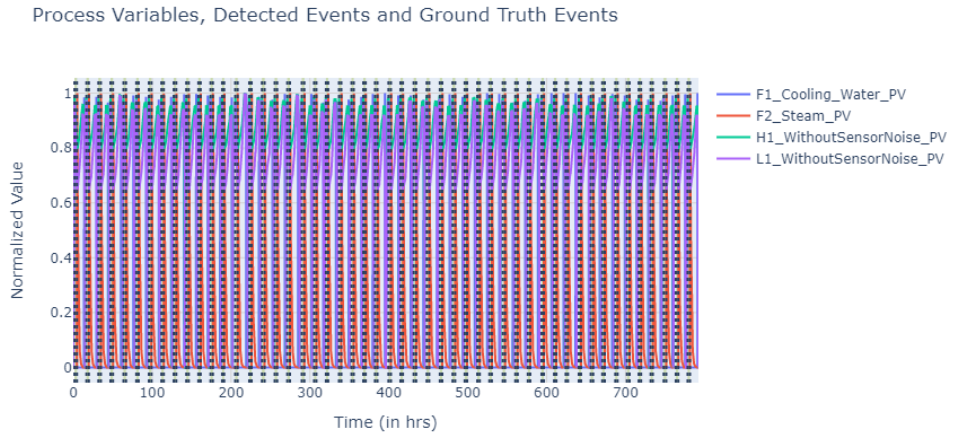

Figure 4.20 Multiple process variables, the ground truth event, the detected change points, and the detected event for all 50 batches in Experiment S2.2.

The following Figure 4.21 shows the multiple process variables, the ground truth event, the detected change points, and the detected event for first 5 batches. The colored traces represent the process variables, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. Due to the overlapping of the ground truth events and the detected events, the green and the orange-colored areas are overlapped and hence, producing the dark green areas. Since these overlapping areas are large, therefore, it can be said that the result in this case is quite good.

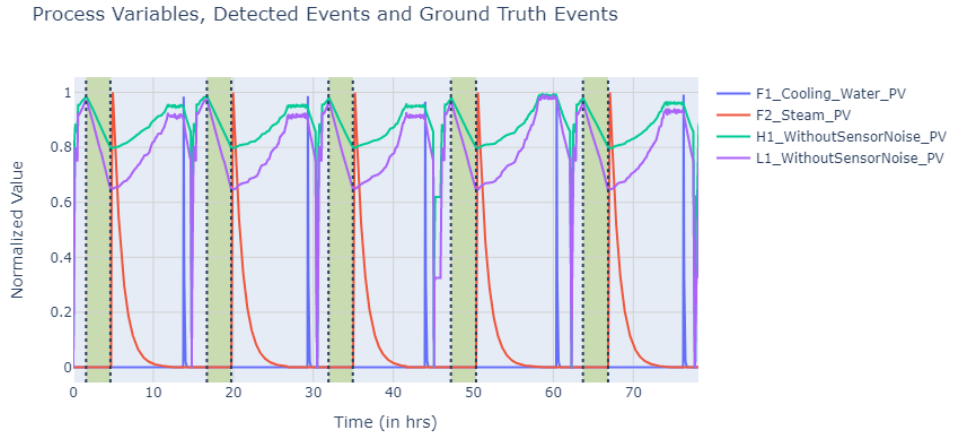

Figure 4.21 Multiple process variables, the ground truth event, the detected change points, and the detected event for first 5 batches in Experiment S2.2.

The following Figure 4.22 shows the evaluation by visualization method for this experiment where all the detected segments are plotted on the same graph together. In this figure, the colored traces represent all the detected segments. From this figure, it can be observed that all the detected segments are very similar. In this experiment, there are 50 ground truth events in 50 batches. Also, 50 detected events and 100 detected change points are obtained from the experiment. So, overall, it can be said that the result from this experiment is quite good.

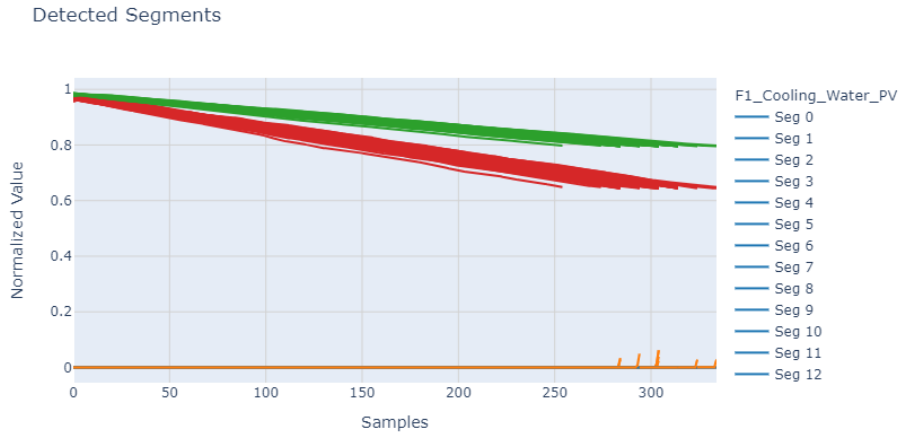

Figure 4.22 Evaluation by visualization method in Experiment S2.2.

- **Experiment S2.3**
  The following Figure 4.23 shows the process variable and the ground truth event from the first batch. In this figure, the blue-colored trace represents the process variable, and the green-colored area represents the ground truth event. This first ground truth event is also taken as the user-selected segment for this experiment.

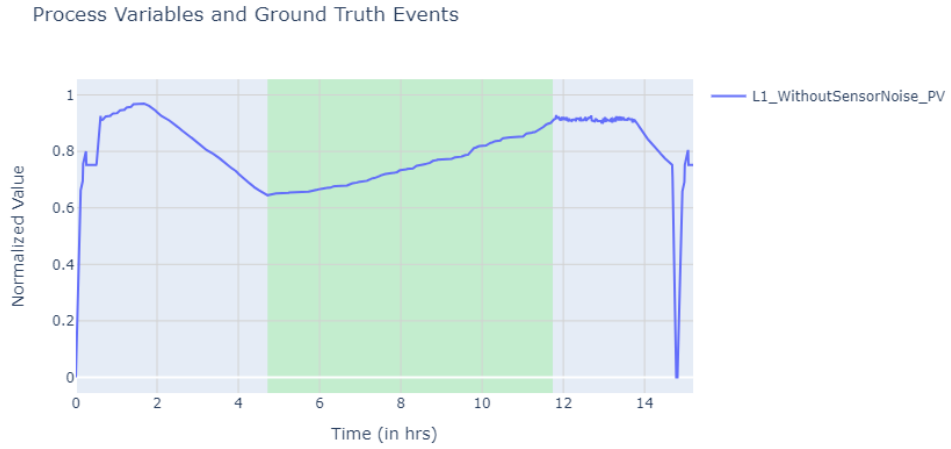

Figure 4.23 Process variable and the ground truth event from the first batch in Experiment S2.3.

The following Figure 4.24 shows the process variable, the ground truth event, the detected change points, and the detected event for all 50 batches. In this figure, the blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. From this figure, it can be observed that the events are detected for all 50 batches. Since the result is not clear here due to the large number of batches, a zoomed-in version of this figure for first 5 batches is shown in the next Figure 4.25.

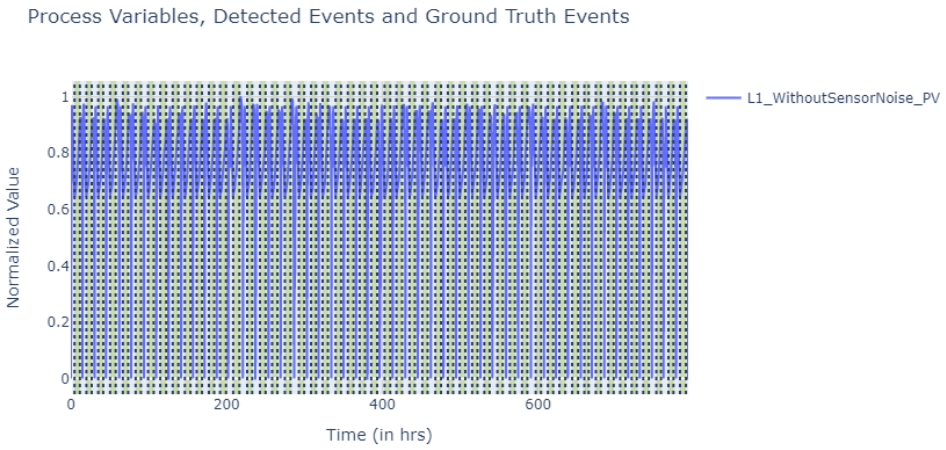

Figure 4.24 Process variable, the ground truth event, the detected change points, and the detected event for all 50 batches in Experiment S2.3.

The following Figure 4.25 shows the process variable, the ground truth event, the detected change points, and the detected event for first 5 batches. The blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. Due to the overlapping of the ground truth events and the detected events, the green and the orange-colored areas are overlapped and hence, producing the dark green areas. Since these overlapping areas are large, therefore, it can be said that the result in this case is quite good.

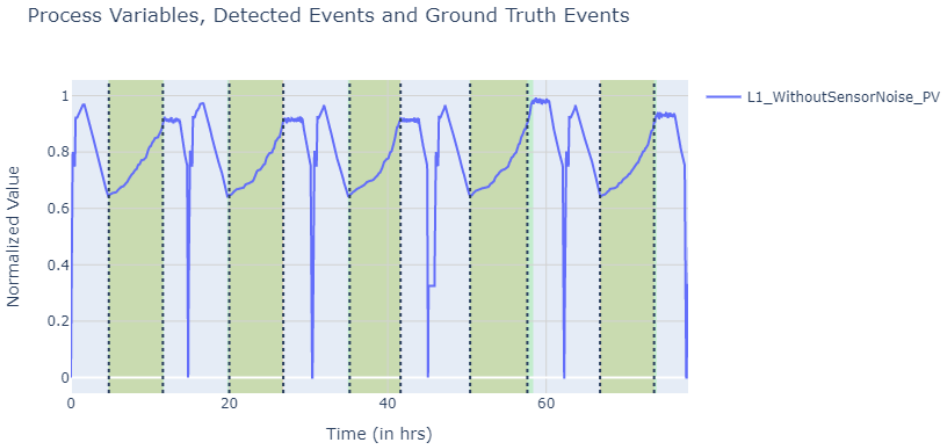

Figure 4.25 Process variable, the ground truth event, the detected change points, and the detected event for first 5 batches in Experiment S2.3.

The following Figure 4.26 shows the evaluation by visualization method for this experiment where all the detected segments are plotted on the same graph together. In this figure, the colored traces represent all the detected segments. From this figure, it can be observed that all the detected segments are very similar. In this experiment, there are 50 ground truth events in 50 batches. Also, 50 detected events and 100 detected change points are obtained from the experiment. So, overall, it can be said that the result from this experiment is quite good.

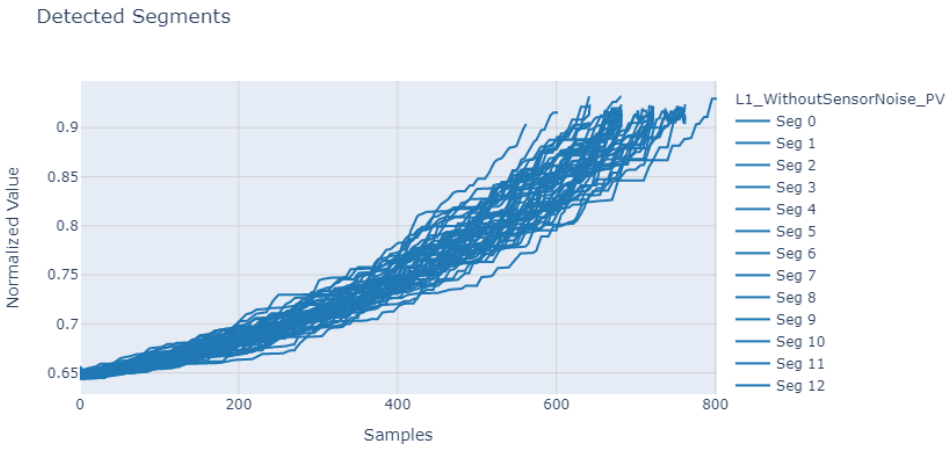

Figure 4.26 Evaluation by visualization method in Experiment S2.3.

- **Experiment S2.4**
  The following Figure 4.27 shows the multiple process variables and the ground truth event from the first batch. In this figure, the colored traces represent the process variables, and the green-colored area represents the ground truth event. This first ground truth event is also taken as the user-selected segment for this experiment.

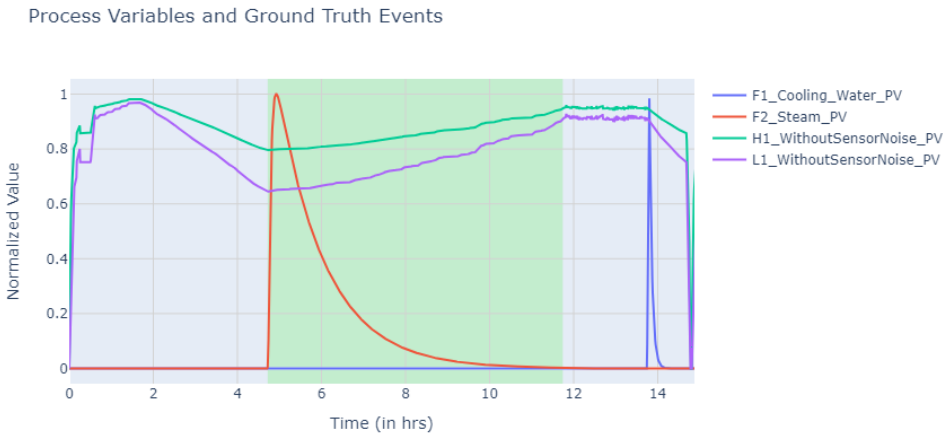

Figure 4.27 Multiple process variables and the ground truth event from the first batch in Experiment S2.4.

The following Figure 4.28 shows the multiple process variables, the ground truth event, the detected change points, and the detected event for all 50 batches. In this figure, the colored traces represent the process variables, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. From this figure, it can be observed that the events are detected for all 50 batches. Since the result is not clear here due to the large number of batches, a zoomed-in version of this figure for first 5 batches is shown in the next Figure 4.29.

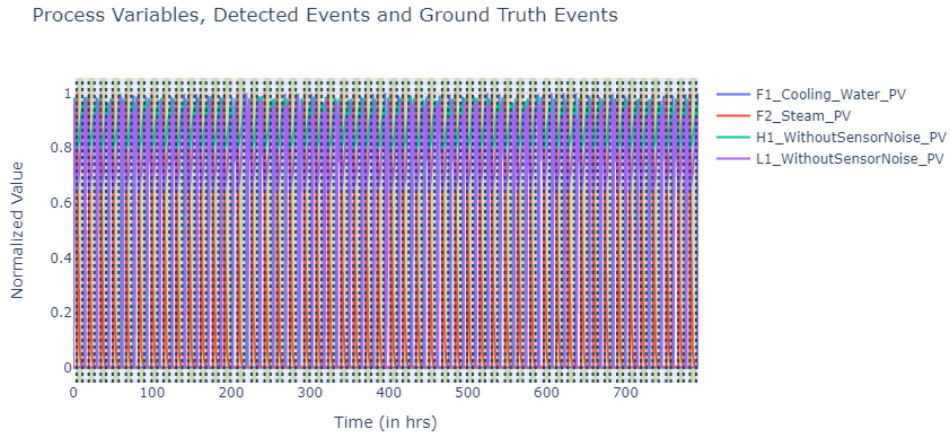

Figure 4.28 Multiple process variables, the ground truth event, the detected change points, and the detected event for all 50 batches in Experiment S2.4.

The following Figure 4.29 shows the multiple process variables, the ground truth event, the detected change points, and the detected event for first 5 batches. The colored traces represent the process variables, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. Due to the overlapping of the ground truth events and the detected events, the green and the orange-colored areas are overlapped and hence, producing the dark green areas. Since these overlapping areas are large, therefore, it can be said that the result in this case is quite good.

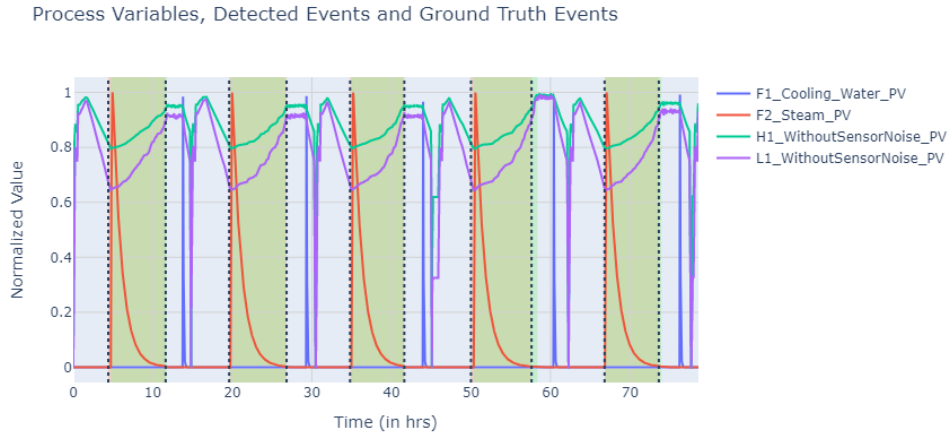

Figure 4.29 Multiple process variables, the ground truth event, the detected change points, and the detected event for first 5 batches in Experiment S2.4.

The following Figure 4.30 shows the evaluation by visualization method for this experiment where all the detected segments are plotted on the same graph together. In this figure, the colored traces represent all the detected segments. From this figure, it can be observed that all the detected segments are very similar. In this experiment, there are 50 ground truth events in 50 batches. Also, 50 detected events and 100 detected change points are obtained from the experiment. So, overall, it can be said that the result from this experiment is quite good.

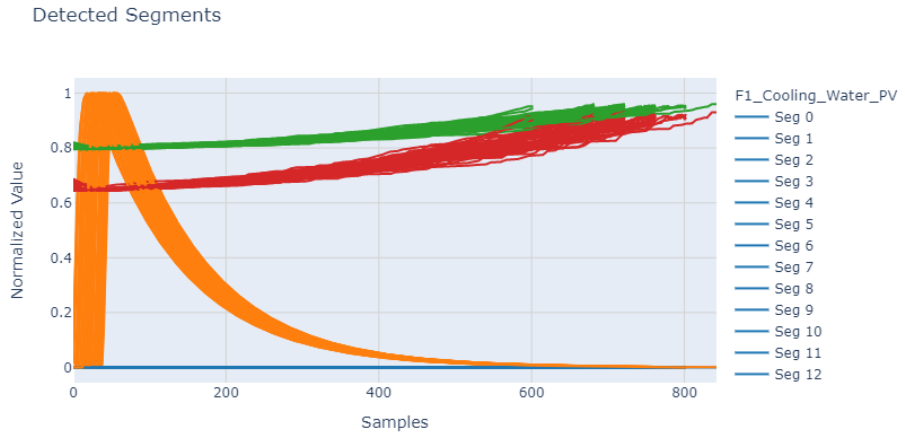

Figure 4.30 Evaluation by visualization method in Experiment S2.4.

- **Experiment S2.5**

  The following Figure 4.31 shows the process variable and the ground truth event from the first batch. In this figure, the blue-colored trace represents the process variable, and the green-colored area represents the ground truth event. This first ground truth event is also taken as the user-selected segment for this experiment. Moreover, the corresponding process variable in this event is very noisy.

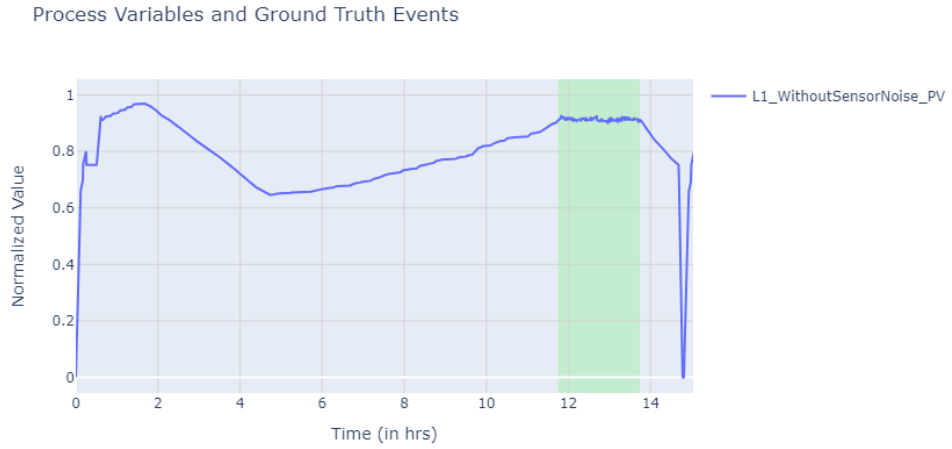

Figure 4.31 Process variable and the ground truth event from the first batch in Experiment S2.5.

The following Figure 4.32 shows the process variable, the ground truth event, the detected change points, and the detected event for all 50 batches. In this figure, the blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. From this figure, it can be observed that the events are detected for 38 batches. Since the result is not clear here due to the large number of batches, a zoomed-in version of this figure for first 5 batches is shown in the next Figure 4.33.

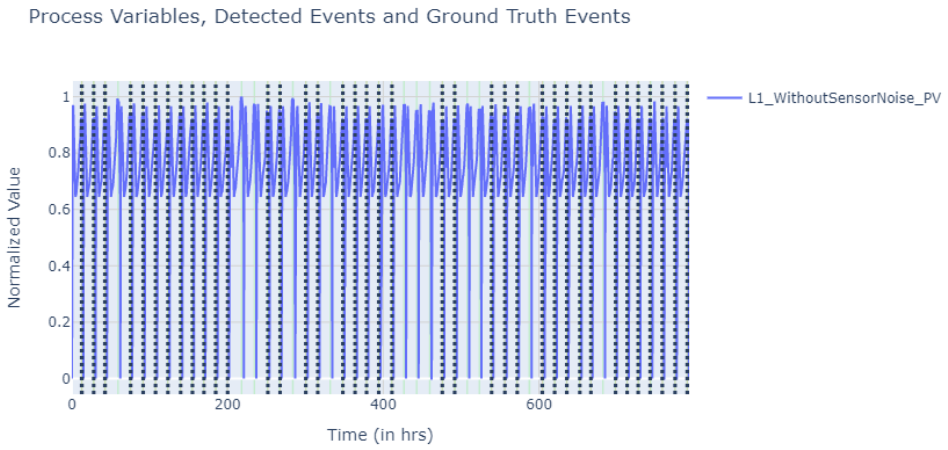

Figure 4.32 Process variable, the ground truth event, the detected change points, and the detected event for all 50 batches in Experiment S2.5.

The following Figure 4.33 shows the process variable, the ground truth event, the detected change points, and the detected event for first 5 batches. The blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. Due to the overlapping of the ground truth events and the detected events, the green and the orange-colored areas are overlapped and hence, producing the dark green areas. It can also be observed that the ground truth events in the batches where the events are not detected are either correspond to the process variable with different shapes due to the noise or the difference in the amplitudes of the process variable is high. So, it can be said that due to the presence of the noisy process variable and the process variable with the large difference in amplitude, the accuracy of the result decreases.

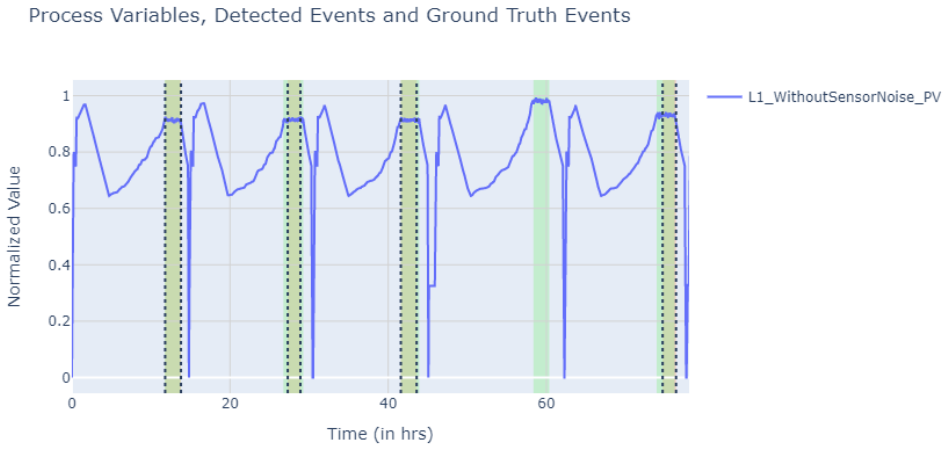

Figure 4.33 Process variable, the ground truth event, the detected change points, and the detected event for first 5 batches in Experiment S2.5.

The following Figure 4.34 shows the evaluation by visualization method for this experiment where all the detected segments are plotted on the same graph together. In this figure, the colored traces represent all the detected segments. From this figure, it can be observed that the detected segments are not very similar due to the presence of the noisy process variable. In this experiment, there are 50 ground truth events in 50 batches. Also, 38 detected events and 76 detected change points are obtained from the experiment. So, overall, it can be said that the accuracy of the result from this experiment is low.

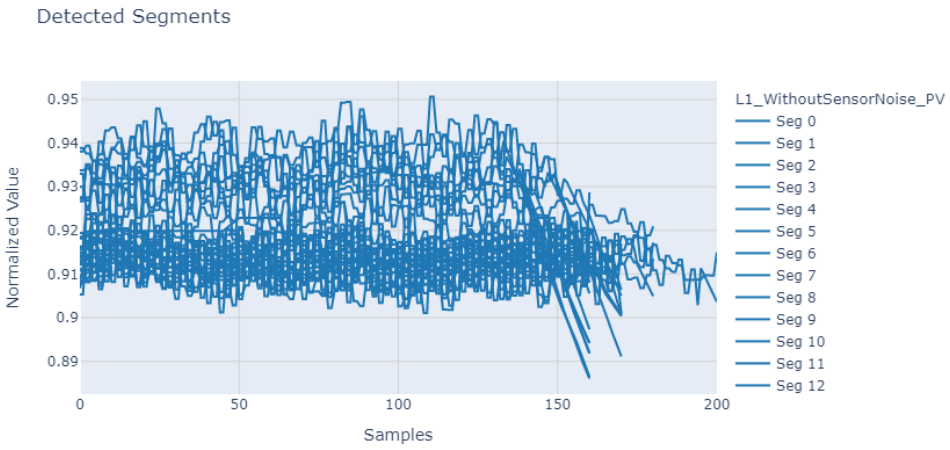

Figure 4.34 Evaluation by visualization method in Experiment S2.5.

- **Experiment S2.6**
  The following Figure 4.35 shows the multiple process variables and the ground truth event from the first batch. In this figure, the colored traces represent the process variables, and the green-colored area represents the ground truth event. This first ground truth event is also taken as the user-selected segment for this experiment. Moreover, the corresponding process variables in this event are very noisy.

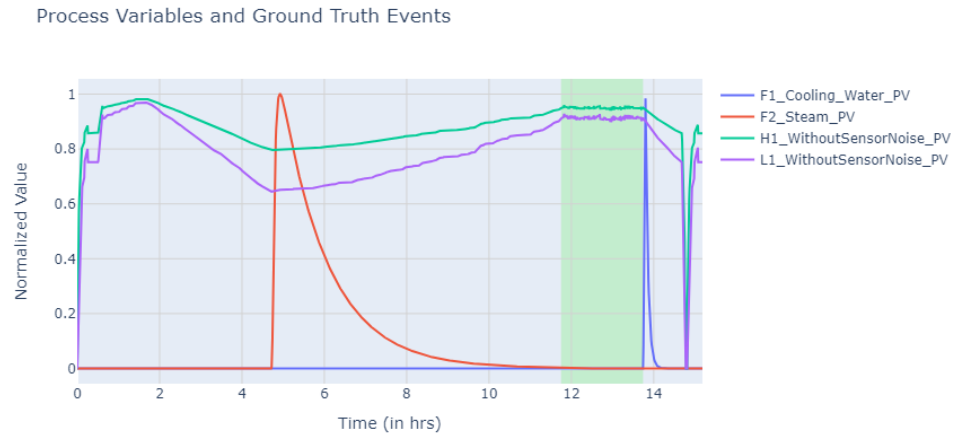

Figure 4.35 Multiple process variables and the ground truth event from the first batch in Experiment S2.6.

The following Figure 4.36 shows the multiple process variables, the ground truth event, the detected change points, and the detected event for all 50 batches. In this figure, the colored traces represent the process variables, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. From this figure, it can be observed that the events are detected for 32 batches. Since the result is not clear here due to the large number of batches, a zoomed-in version of this figure for first 5 batches is shown in the next Figure 4.37.

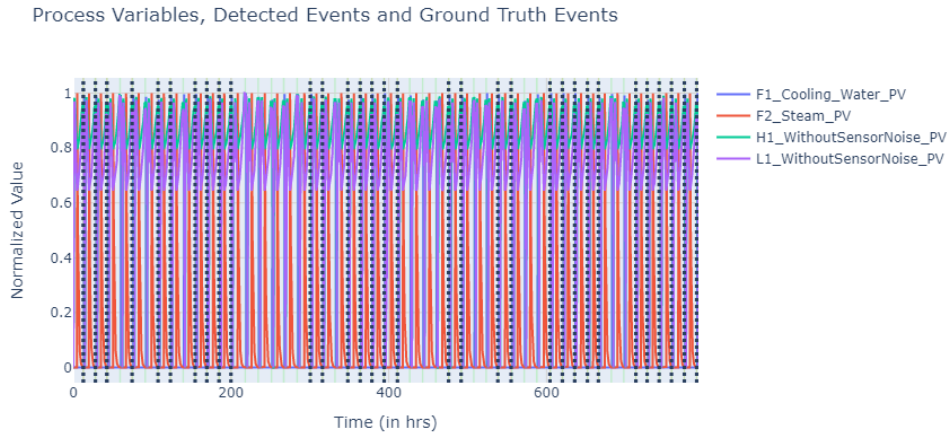

Figure 4.36 Multiple process variables, the ground truth event, the detected change points, and the detected event for all 50 batches in Experiment S2.6.

The following Figure 4.37 shows the multiple process variables, the ground truth event, the detected change points, and the detected event for first 5 batches. The colored traces represent the process variables, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. Due to the overlapping of the ground truth events and the detected events, the green and the orange-colored areas are overlapped and hence, producing the dark green areas. It can also be observed that the ground truth events in the batches where the events are not detected are either corresponding to the process variables with different shapes due to the noise or the difference in the amplitudes of the process variables is high. So, it can be said that due to the presence of the noisy process variables and the process variables with the large difference in amplitude, the accuracy of the result decreases.

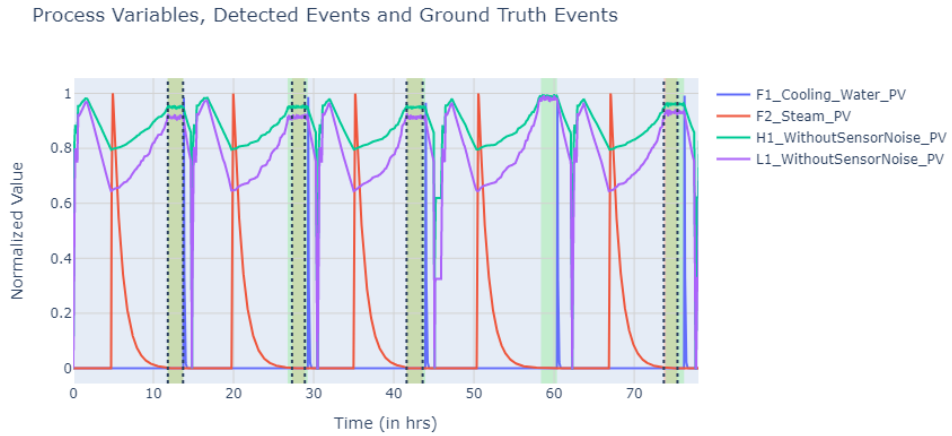

Figure 4.37 Multiple process variables, the ground truth event, the detected change points, and the detected event for first 5 batches in Experiment S2.6.

The following Figure 4.38 shows the evaluation by visualization method for this experiment where all the detected segments are plotted on the same graph together. In this figure, the colored traces represent all the detected segments. From this figure, it can be observed that the detected segments are not very similar due to the presence of the noisy process variable. In this experiment, there are 50 ground truth events in 50 batches. Also, 32 detected events and 64 detected change points are obtained from the experiment. So, overall, it can be said that the accuracy of the result from this experiment is low.

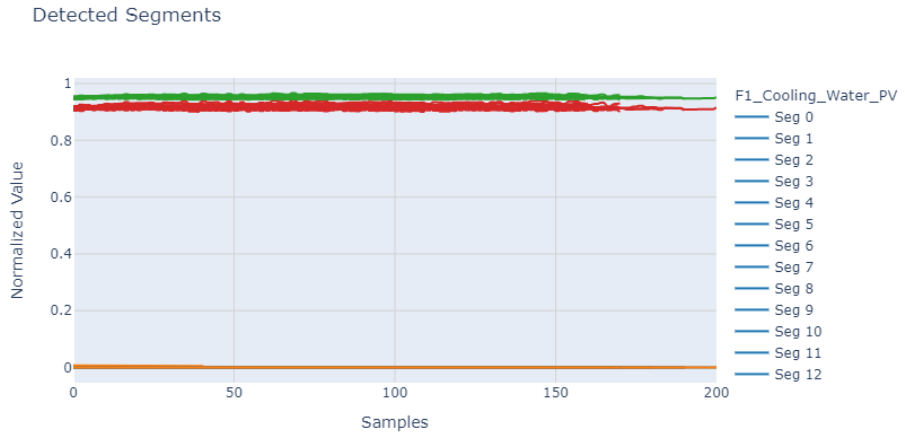

Figure 4.38 Evaluation by visualization method in Experiment S2.6.

- **Experiment S2.7**
  The following Figure 4.39 shows the process variable and the ground truth event from the first batch. In this figure, the blue-colored trace represents the process variable, and the green-colored area represents the ground truth event. This first ground truth event is also taken as the user-selected segment for this experiment. Moreover, this event is a complex event due to the absence of the relevant process variable and also due to large variations in the duration of the event in different batches. The absence of the relevant process variable means that the corresponding process variable in this event has different behavior/shape/characteristics in different batches.

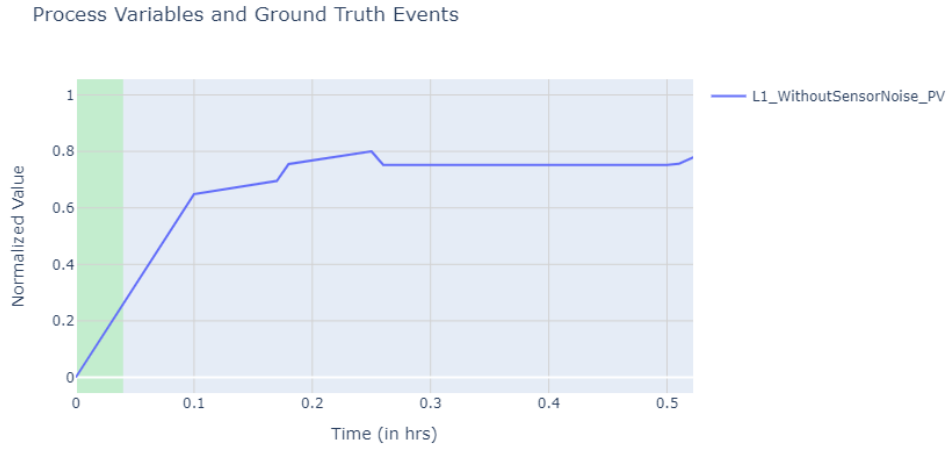

Figure 4.39 Process variable and the ground truth event from the first batch in Experiment S2.7.

The following Figure 4.40 shows the process variable, the ground truth event, the detected change points, and the detected event for all 50 batches. In this figure, the blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. From this figure, it can be observed that the events are detected for all 50 batches. Since the result is not clear here due to the large number of batches, a zoomed-in version of this figure for first and sixth batch are shown in the next two figures Figure 4.41 and Figure 4.42.

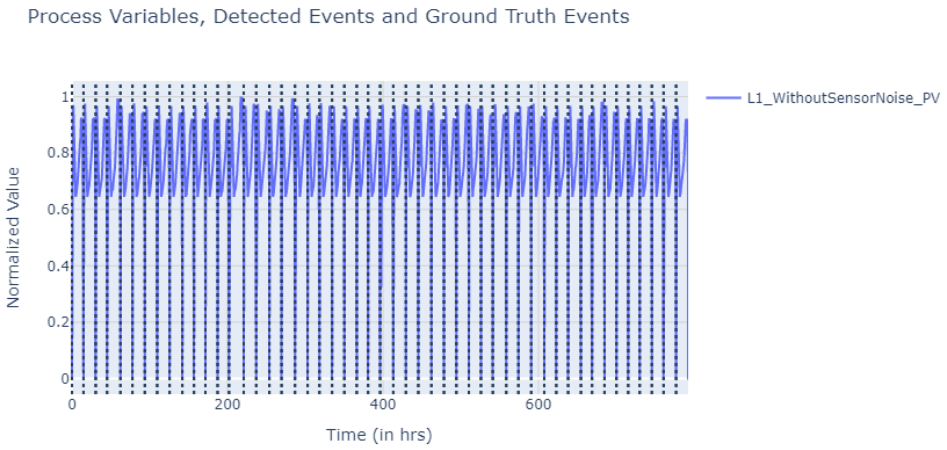

Figure 4.40 Process variable, the ground truth event, the detected change points, and the detected event for all 50 batches in Experiment S2.7.

The following Figure 4.41 shows the process variable, the ground truth event, the detected change points, and the detected event for the first batch. The blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored area represents the ground truth event, and the orange-colored area represents the detected event. Due to the overlapping of the ground truth event and the detected event, the green and the orange-colored areas are overlapped and hence, producing the dark green area. Since the overlapping area in the first batch is large, therefore, it can be said that the result for the first batch is good.

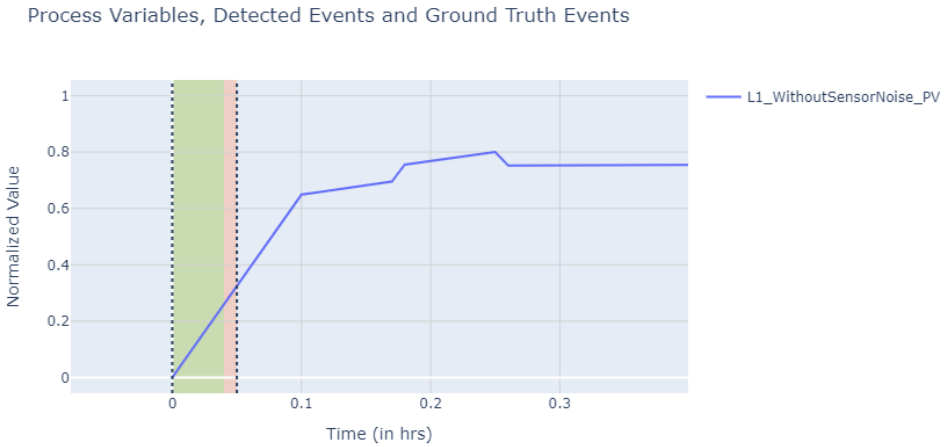

Figure 4.41 Process variable, the ground truth event, the detected change points, and the detected event for the first batch in Experiment S2.7.

The following Figure 4.42 shows the process variable, the ground truth event, the detected change points, and the detected event for the sixth batch. The blue-colored trace represents the process variable, vertical dashed lines represent the detected change points, the green-colored area represents the ground truth event, and the orange-colored area represents the detected event. Due to the overlapping of the ground truth event and the detected event, the green and the orange-colored areas are overlapped and hence, producing the dark green area. Since the overlapping area in the sixth batch is small, therefore, it can be said that the result for the sixth batch is poor.

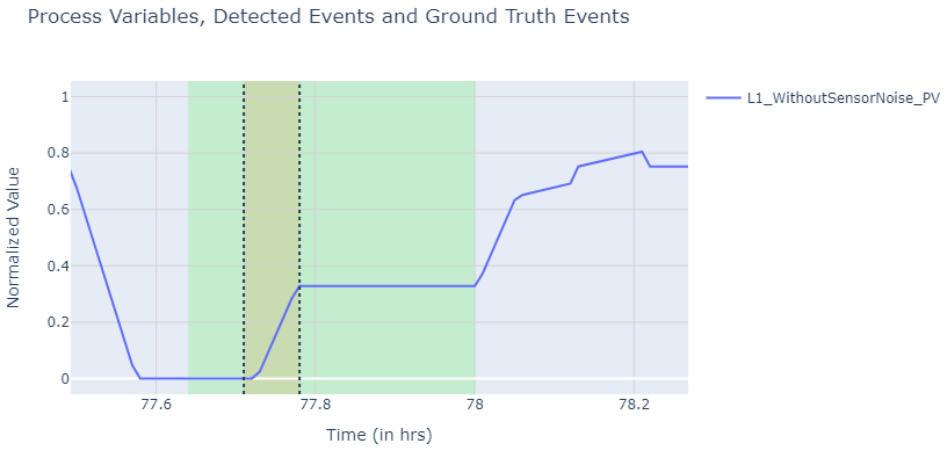

Figure 4.42 Process variable, the ground truth event, the detected change points, and the detected event for the sixth batch in Experiment S2.7.

The following Figure 4.43 shows the evaluation by visualization method for this experiment where all the detected segments are plotted on the same graph together. In this figure, the colored traces represent all the detected segments. From this figure, it can be observed that all the detected segments are very similar. In this experiment, there are 50 ground truth events in 50 batches. Also, 50 detected events and 100 detected change points are obtained from the experiment. Although the events are detected for all the batches and the detected segments are also similar, the overlapping is good only for some batches due to the absence of the relevant process variable in this complex event. So, overall, it can be said that the accuracy of the result from this experiment is low.

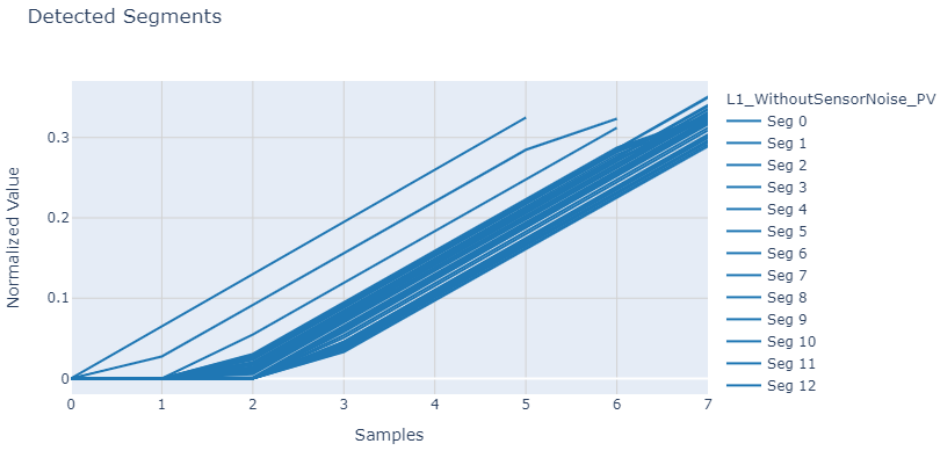

Figure 4.43 Evaluation by visualization method in Experiment S2.7.

- **Experiment S2.8**
  The following Figure 4.44 shows the multiple process variables and the ground truth event from the first batch. In this figure, the colored traces represent the process variables, and the green-colored area represents the ground truth event. This first ground truth event is also taken as the user-selected segment for this experiment. Moreover, this event is a complex event due to the absence of the relevant process variables and also due to large variations in the duration of the event in different batches. The absence of the relevant process variables means that the corresponding process variables in this event have different behavior/shape/characteristics in different batches.

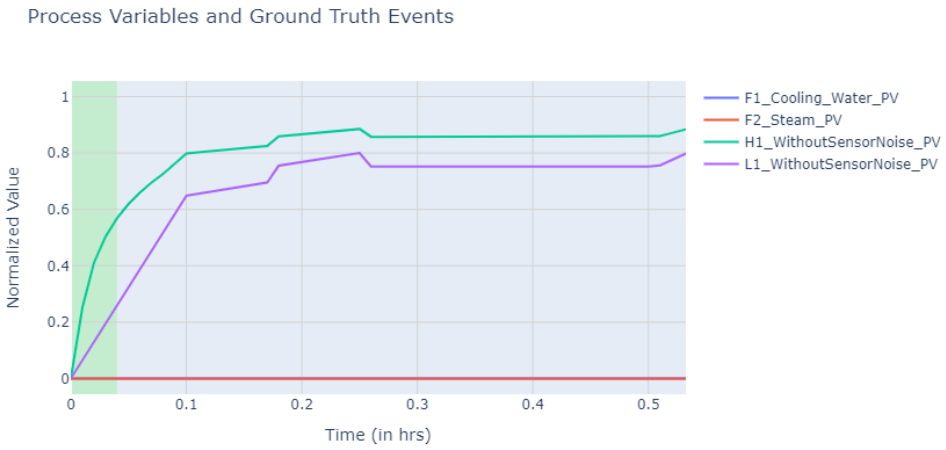

Figure 4.44 Multiple process variables and the ground truth event from the first batch in Experiment S2.8.

The following Figure 4.45 shows the multiple process variables, the ground truth event, the detected change points, and the detected event for all 50 batches. In this figure, the colored traces represent the process variables, vertical dashed lines represent the detected change points, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. From this figure, it can be observed that the events are detected for all 50 batches. Since the result is not clear here due to the large number of batches, a zoomed-in version of this figure for first and sixth batch are shown in the next two figures Figure 4.46 and Figure 4.47.

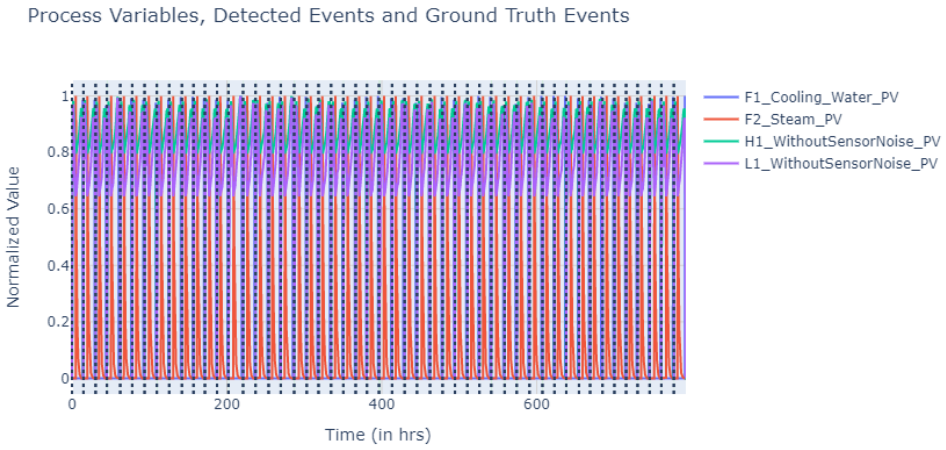

Figure 4.45 Multiple process variables, the ground truth event, the detected change points, and the detected event for all 50 batches in Experiment S2.8.

The following Figure 4.46 shows the multiple process variables, the ground truth event, the detected change points, and the detected event for the first batch. The colored traces represent the process variables, vertical dashed lines represent the detected change points, the green-colored area represents the ground truth event, and the orange-colored area represents the detected event. Due to the overlapping of the ground truth event and the detected event, the green and the orange-colored areas are overlapped and hence, producing the dark green area. Since the overlapping area in the first batch is large, therefore, it can be said that the result for the first batch is good.

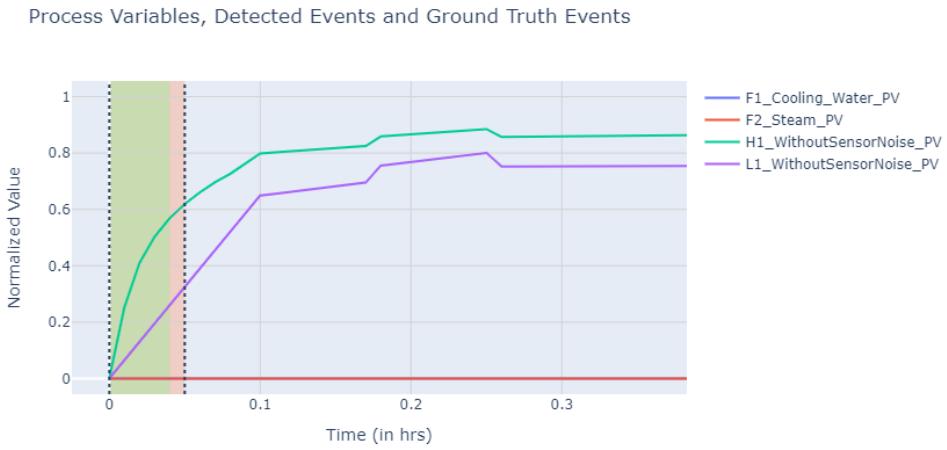

Figure 4.46 Multiple process variables, the ground truth event, the detected change points, and the detected event for the first batch in Experiment S2.8.

The following Figure 4.47 shows the multiple process variables, the ground truth event, the detected change points, and the detected event for the sixth batch. The colored traces represent the process variables, vertical dashed lines represent the detected change points, the green-colored area represents the ground truth event, and the orange-colored area represents the detected event. Due to the overlapping of the ground truth event and the detected event, the green and the orange-colored areas are overlapped and hence, producing the dark green area. Since the overlapping area in the sixth batch is small, therefore, it can be said that the result for the sixth batch is poor.

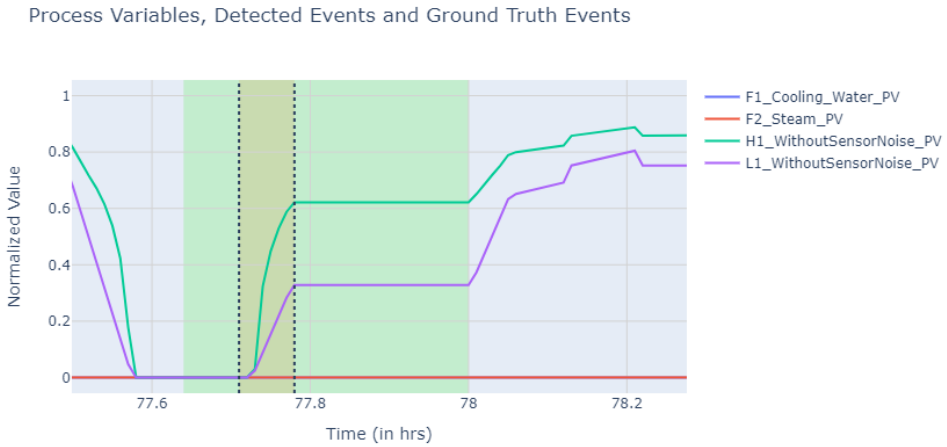

Figure 4.47 Multiple process variables, the ground truth event, the detected change points, and the detected event for the sixth batch in Experiment S2.8.

The following Figure 4.48 shows the evaluation by visualization method for this experiment where all the detected segments are plotted on the same graph together. In this figure, the colored traces represent all the detected segments. From this figure, it can be observed that all the detected segments are very similar. In this experiment, there are 50 ground truth events in 50 batches. Also, 50 detected events and 100 detected change points are obtained from the experiment. Although the events are detected for all the batches and the detected segments are also similar, the overlapping is good only for some batches due to the absence of the relevant process variable in this complex event. So, overall, it can be said that the accuracy of the result from this experiment is low.

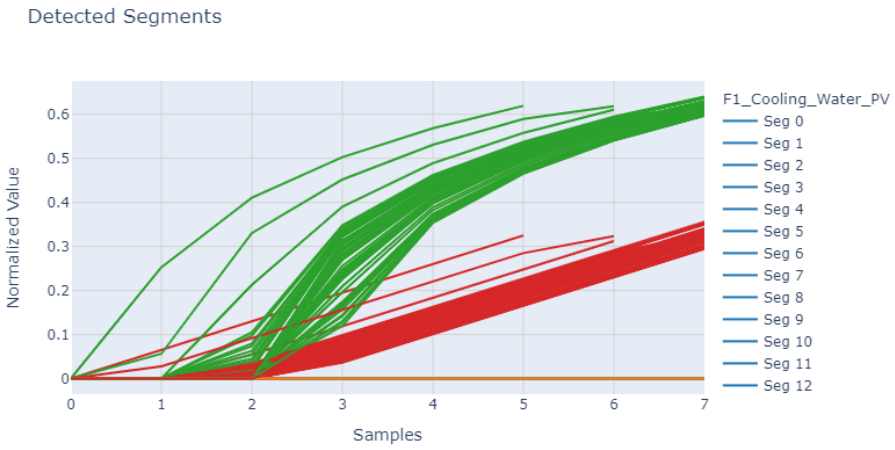

Figure 4.48 Evaluation by visualization method in Experiment S2.8.

The following Table 4.5 demonstrates the summary of results from the second set of experiments. In this table, the columns represent the name of the experiments, the event for which the experiments are performed, the experiments with the univariate/multivariate case, the values from the used evaluation metrics, and the time taken to compute the results respectively. Moreover, the percentage in start location error, end location error, and duration error are calculated over the mean of true event duration.

Table 4.5 Summary of results from the second set of experiments.

Experiment	Event Name	Univariate / Multivariate	True Event Duration Information		Detected Event Duration Information		Annotation Error = | N(true events) − N(detected events) |	Start location error (between true and detected events)		End location error (between true and detected events)		Duration error (between true and detected events)		Computation Time
			Mean [hrs]	Std [hrs]	Mean [hrs]	Std [hrs]		[hrs]	[%]	[hrs]	[%]	[hrs]	[%]	
Experiment S2.1	Product Transfer	Univariate	3.0128	0.1484	3.062	0.1539	0	0.0382	1.27	0.0406	1.35	0.0608	2.02	1m 41.7s
Experiment S2.2		Multivariate	3.0128	0.1484	3.002	0.1522	0	0.0382	1.27	0.0286	0.95	0.0476	1.58	2m 9.4s
Experiment S2.3	Reaction	Univariate	7.2574	0.4598	6.9399	0.44	0	0.139	1.92	0.2996	4.13	0.3646	5.02	2m 22.6s
Experiment S2.4		Multivariate	7.2574	0.4598	7.236	0.4593	0	0.1913	2.64	0.2972	4.09	0.2726	3.76	3m 25.6s
Experiment S2.5	Post Reaction	Univariate	2.0972	0.1649	1.6711	0.1024	12	0.3587	17.1	0.2663	12.69	0.3542	16.89	1m 6.4s
Experiment S2.6		Multivariate	2.0972	0.1649	1.65	0.0935	18	0.2169	10.34	0.3953	18.85	0.3599	17.16	1m 50.9s
Experiment S2.7	Add Educt 1	Univariate	0.3892	0.3081	0.0692	0.0034	0	0.055	14.13	0.2702	69.42	0.3212	82.53	35.9s
Experiment S2.8		Multivariate	0.3892	0.3081	0.0692	0.0034	0	0.055	14.13	0.2706	69.53	0.3212	82.53	54.9s

The results from the second set of experiments show that the proposed interactive CPD algorithm with the VLSW search method and the DTW distance as the cost function for events identification has achieved good performance for two events, namely Product Transfer and Reaction. There is no annotation error, indicating that the number of detected events matches exactly with the number of events. The duration error and the location errors are also relatively small. On the other hand, the performance for the events Post Reaction and Add Educt 1 are not as good as the performance on the other events. Although the annotation error for the event Add Educt 1 is 0, the location errors and the duration error are very large. The reason for this is the event Add Educt 1 is a complex event due to the absence of the relevant process variables and also due to large variation in the duration of the event in different batches. The absence of the relevant process variables means that the corresponding process variables in this event have different behavior/shape/characteristics in different batches. The results are more accurate for simple events having relevant process variables than complex events without relevant process variables. Since the interactive CPD algorithm always finds the segments containing process variables with similar behavior/characteristics/shape, this will work for the simple events with relevant process variables, but this will not work for the complex events without relevant process variables. However, it could be possible that the complex event may corresponds to different sets of relevant process variables in different batches, and therefore, it's worthwhile investigating that if repeating the interactive CPD algorithm multiple times for the same complex event by identifying different segments of relevant process variables each time for the complete dataset will work or not. Furthermore, for the Post Reaction event, the annotation error, the location errors, and the duration error are large. This is because the Post Reaction event corresponds to the noisy process variables. The accuracy decreases in the case of events having noisy process variables as the shape of these variables changes due to the noise. The accuracy also decreases when

the process variables in an event across different batches have a similar shape but there is a large difference in their amplitudes because DTW distance is sensitive to the shape as well as the amplitude. However, other cost functions in place of DTW distance could be applied like global alignment kernel (GAK) and autoencoder. Moreover, the results in the multivariate case are more accurate than that of the univariate case except for the case in which the event corresponds to the noisy process variables. The accuracy of the results in the multivariate case further decreases as compared to the univariate case in the event with noisy process variables because the multivariate DTW distance increases due to the multiple noisy process variables. Additionally, the computation time is more for the longer duration events than that for the shorter duration events. The computation time also increases with the increase in the length of the user-selected segment as well as with the increase in the minimum and the maximum deviation parameters, but it decreases with the increase in the steps parameters. The decrease in the steps parameter increases the accuracy of the results but also increases the computation time.

4.2 Experiments on Real-life Dataset

In this section, the real-life dataset is used for the experiments with interactive change point detection. This real-life dataset contains the private process industry data. Therefore, the names and the labels of the process variables and the events of the real-life dataset are anonymized.

In the real-life dataset that is used for the experiments, there are missing values. So, in the preprocessing steps after normalization, the complete dataset with missing values in between is divided into multiple chunks of continuous-time datasets with equidistant sampling and without any missing values in between. This division is done to avoid the unnecessary detections of change points and to speed up the computation. The following Figure 4.49 illustrates the preprocessed process variables and the event from multiple batches of the real-life dataset. The colored traces represent the process variables, the green-colored area represents the event, and the grey-colored area represents the missing values.

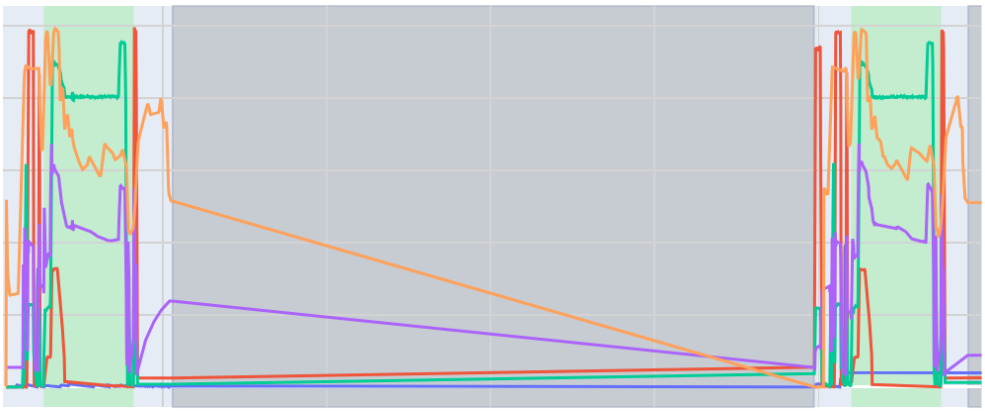

Figure 4.49 Preprocessed process variables and the event from multiple batches of the real-life dataset.

4.2.1 Experiments Setup

In the experiments with interactive change point detection for events identification on the real-life dataset, varying length sliding window (VLSW) search method, dynamic time warping (DTW) distance as the customized cost function, and VLSW search method parameters as the tuning parameters are used. More details of these methods and functions can be found in section 3.2.

For events identification with interactive change point detection, a set of experiments is performed here. The following Table 4.6 describes this set of experiments. In this table, the columns represent the name of the experiment set, the description of the experiment set, the dataset with number of batches used as well as whether the univariate/multivariate data is used for the experiment set, the events for which the experiments in the experiment set are performed, the search method used, the customized cost function used, the tuning parameters used, and the evaluation metrics/methods used respectively.

Table 4.6 Description of the experiment set performed with interactive change point detection for events identification on real-life dataset.

S. No.	Experiment Set	Description	Dataset	Events	Search method	Customized cost functions	Tuning parameters	Evaluation
1.	Experiment R1	To get the interactive CPD results for multiple batch data with the VLSW search method and the DTW distance as the customized cost function by tuning different parameters using user feedback to get artificial labels for the events accurately.	Real-life Dataset (45 batches of data; both univariate and multivariate)	• Event 1 • Event 2 • Event 3 • Event 4 • Event 5	VLSW	DTW distance	VLSW search method parameters	• Annotation error • Mean of true event duration • Standard deviation of true event duration • Mean of detected event duration • Standard deviation of detected event duration • Start location error • End location error • Duration error • Evaluation by visualization

This set of experiments contains several experiments composed by taking the different events. The detailed description of this set of experiments are following:

Experiment R1

The aim of this set of experiments is to get the interactive CPD results for multiple batch data with the VLSW search method and the DTW distance as the customized cost function by tuning different parameters using user feedback to get artificial labels for the events accurately. The dataset used for these experiments is the real-life dataset containing 45 batches of data. Moreover, both univariate and multivariate data are used in different experiments of this set. For search method, VLSW is used; for customized cost function, DTW distance is used; for tuning parameters, VLSW search method parameters are used which are minimum deviation, maximum deviation, steps, cost threshold, and minimum segment distance. More details on this search method, the customized cost function, and the tuning parameters can be found in sub-sections 3.2.6, 3.2.5, and 3.2.8 respectively. Along with these, different evaluation metrics/method are also used for evaluating the results. These evaluation metrics and method are Annotation error, Mean of true event duration, Standard deviation of true event duration, Mean of detected event duration, Standard deviation of detected event duration, Start location error, End location error, Duration error, and Evaluation by visualization method. All detailed descriptions of these evaluation metrics and method can be found in sub-section 3.2.7.

For all the experiments in this set, the steps followed are similar to that of Experiment S2 in sub-section 4.1.1. The only difference is in the preprocessing steps and the way in which the results are obtained. Due to the presence of the missing values in the real-life dataset, one further step is performed in the preprocessing steps. After normalization, the complete dataset with missing values in between is divided into multiple chunks of continuous-time datasets with equidistant sampling and without any missing values in between. This division is done to avoid the unnecessary detections of change points and to speed up the computation. For obtaining the results, the tuning parameters along with the user-selected segment and the continuous chunks of the dataset (one at a time) are passed into the interactive CPD algorithm with the VLSW search method and the DTW distance as the cost function to get the initial detected change point indices for each chunk. After getting the initial detected change point indices for each chunk, these indices are combined to get the initial change point indices for the complete dataset.

The number of experiments performed under this set of experiments is 5. The following Table 4.7 describes these 5 experiments. In this table, the columns represent the name of the experiments, the dataset with number of batches used as well as whether the univariate/multivariate data is used for the experiments, the event for which the experiments are performed, the process variable(s) used for the experiments (single process variable in the univariate case and multiple process variables in the multivariate case), the search method used, the customized cost function used, the values from the used tuning parameters respectively.

Table 4.7 Description of experiments under the experiment set.

S. No.	Experiment	Dataset	Event	Used process variable(s)	Search method	Customized cost function	Tuning parameters
1.	Experiment R1.1	Real-time Dataset (45 batches of data; uni-variate)	Event 1	PV1	VLSW	DTW distance	• Minimum deviation = 0.2 • Maximum deviation = 0.2 • Steps = 5 • Cost threshold = 1 • Minimum distance = 0.5; Minimum segment distance = 268
2.	Experiment R1.2	Real-time Dataset (45 batches of data; uni-variate)	Event 2	PV1	VLSW	DTW distance	• Minimum deviation = 0.2 • Maximum deviation = 0.2 • Steps = 5 • Cost threshold = 1 • Minimum distance = 0.5; Minimum segment distance = 915
3.	Experiment R1.3	Real-time Dataset (45 batches of data; multi-variate)	Event 3	• PV2 • PV3 • PV4	VLSW	DTW distance	• Minimum deviation = 0.8 • Maximum deviation = 0.1 • Steps = 1 • Cost threshold = 1.5 • Minimum distance = 0.5; Minimum segment distance = 264
4.	Experiment R1.4	Real-time Dataset (45 batches of data; multi-variate)	Event 4	• PV5 • PV6 • PV7 • PV8 • PV9	VLSW	DTW distance	• Minimum deviation = 0.8 • Maximum deviation = 0.4 • Steps = 5 • Cost threshold = 3.8 • Minimum distance = 0.5; Minimum segment distance = 942
5.	Experiment R1.5	Real-time Dataset (45 batches of data; multi-variate)	Event 5	• PV2 • PV9	VLSW	DTW distance	• Minimum deviation = 0.8 • Maximum deviation = 3 • Steps = 1 • Cost threshold = 1.6 • Minimum distance = 1; Minimum segment distance = 2375

4.2.2 Results

There is a set of experiments that are performed with interactive change point detection for the identification of the events on the real-life dataset. The description of this set of experiments is described in the experiments setup sub-section 4.2.1 and the results are presented in this sub-section.

Results of Experiment R1

There are 5 experiments in this set of experiments. The result of Experiment R1.4 is presented here. The following Figure 4.50 shows the multiple process variables, the ground truth event, the detected change points, and the detected event from multiples batches of the real-life dataset. The colored traces represent the process variables, vertical dashed lines represent the detected change points, the grey colored areas represent the missing values, the green-colored areas represent the ground truth events and the orange-colored areas represent the detected events. Due to the overlapping of the ground truth events and the detected events, the green and the orange-colored areas

are overlapped and hence, producing the dark green areas. Since these overlapping areas are large, therefore, it can be said that the result in this case is quite good.

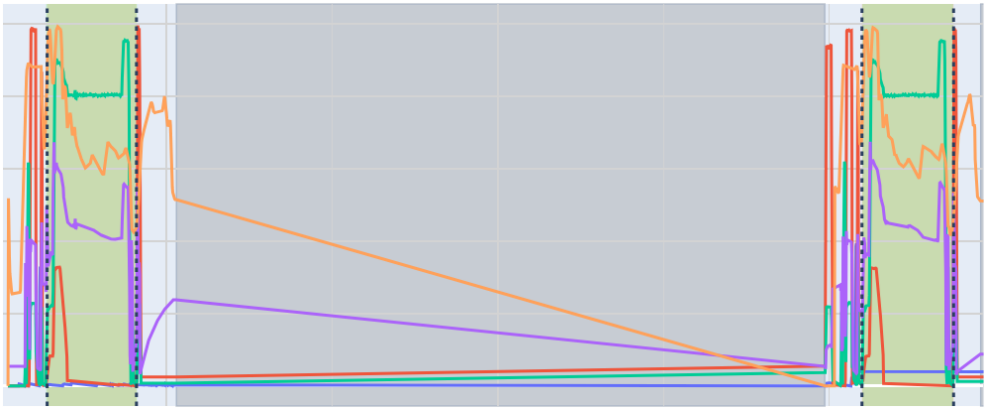

Figure 4.50 Multiple process variables, the ground truth event, the detected change points, and the detected event from multiples batches of the real-life dataset in Experiment R1.4.

The following Figure 4.51shows the evaluation by visualization method for this experiment where all the detected segments are plotted on the same graph together. In this figure, the colored traces represent all the detected segments. From this figure, it can be observed that all the detected segments are very similar. In this experiment, there are 45 ground truth events in 45 batches. Also, 45 detected events and 90 detected change points are obtained from the experiment. So, overall, it can be said that the result from this experiment is quite good.

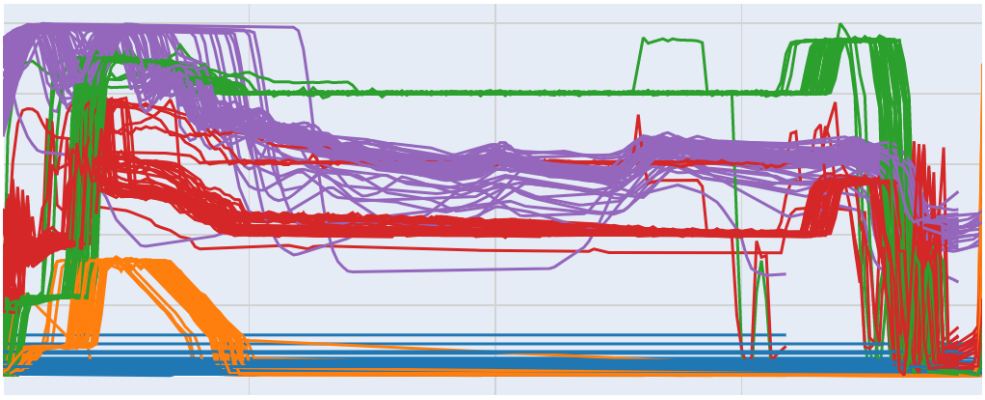

Figure 4.51 Evaluation by visualization method in Experiment R1.4.

The following Table 4.8 demonstrates the summary of results from the set of experiments on the real-life dataset. In this table, the columns represent the name of the experiments, the event for which the experiments are performed, the experiments with the univariate/multivariate case, the values from the used evaluation metrics, and the time taken to compute the results respectively. Moreover, the percentage in start location error, end location error, and duration error are calculated over the mean of true event duration.

Table 4.8 Summary of results from the set of experiments on the real-life dataset.

Experiment	Event Name	Univariate / Multivariate	True Event Duration Information		Detected Event Duration Information		Annotation Error = | N(true events) – N(detected events) |	Start location error (between true and detected events)		End location error (between true and detected events)		Duration error (between true and detected events)		Computation Time
			Mean [hrs:min:sec]	Std [hrs:min:sec]	Mean [hrs:min:sec]	Std [hrs:min:sec]		[hrs:min:sec]	[%]	[hrs:min:sec]	[%]	[hrs:min:sec]	[%]	
Experiment R1.1	Event 1	Univariate	03:03:05	00:00:47	03:08:47	00:04:37	0	00:04:21	2.38	00:02:35	1.41	00:06:06	3.33	10.7s
Experiment R1.2	Event 2	Univariate	05:14:34	00:25:26	05:04:26	00:21:45	0	00:08:27	2.69	00:16:35	5.27	00:14:07	4.49	19.8s
Experiment R1.3	Event 3	Multivariate	00:13:53	00:14:35	00:14:33	00:03:40	1	00:01:38	11.76	00:06:52	49.46	00:05:57	42.86	23.2s
Experiment R1.4	Event 4	Multivariate	03:17:13	00:00:13	03:13:07	00:07:06	0	00:02:36	1.32	00:03:33	1.8	00:05:14	2.65	38.5s
Experiment R1.5	Event 5	Multivariate	00:18:43	00:23:43	00:18:05	00:14:04	22	00:14:32	77.65	00:08:43	46.57	00:03:24	18.17	1m 17.9s

The results from the set of experiments on the real-life dataset show that the proposed interactive CPD algorithm with the VLSW search method and the DTW distance as the cost function for events identification has achieved good performance for three events, namely Event 1, Event 2, and Event 4. There is no annotation error, indicating

that the number of detected events matches exactly with the number of events. The duration error and the location errors are also relatively small. On the other hand, the performance for the events Event 3 and Event 5 is not as good as the performance on the other events. Although the annotation error for Event 3 is only 1, the location errors and the duration error are very large. The reason for this is Event 3 is a complex event due to the absence of the relevant process variables and also due to large variation in the duration of the event in different batches. Similarly, for Event 5, the process variables in this complex event are so irrelevant that along with the location errors and the duration error, the annotation error is also very large. More detailed explanation of these reasons can be found in the results for Experiment S2 in sub-section 4.1.2. Additionally, the computation time is more for the longer duration events than that for the shorter duration events. The computation time also increases with the increase in the length of the user-selected segment as well as with the increase in the minimum and the maximum deviation parameters, but it decreases with the increase in the steps parameters. The decrease in the steps parameter increases the accuracy of the results but also increases the computation time.

5 Discussion

In this chapter, the summary of general findings from the experiments with the interactive change point detection for the events identification on different datasets along with their results and future works are discussed. There are two sections in this chapter: the summary of general findings from the experiments and the results are discussed in the first section and then, different future works are discussed in the second section.

5.1 Experiments and Results

The following are the general findings from the experiments and results:

- The interactive CPD results are more accurate than the unsupervised CPD results due to the introduction of user feedback in the interactive CPD approach.
- In the experiments with the interactive CPD using the PELT search method, for the customized cost function, the DTW distance performed better than the prediction errors from the linear (linear regression) and non-linear (polynomial regression with degree 2) trained model. Furthermore, for the customized cost function, the performance of the prediction error from the non-linear model was slightly better than that of the linear model due to the non-linear characteristics of the process variable in the event.
- The performance of the interactive CPD for the events identification using the VLSW search method with DTW distance as the cost function was comparable with that of the PELT search method with DTW distance as the cost function except for the results obtained for the first and last batch in case of the PELT search method. The result obtained for the first and last batch in the case of the PELT search method was not accurate. However, in this case, to improve the result from the first and the last batch, zero padding at the start and at the end of the dataset can be applied.
- The results are more accurate for simple events having relevant process variables than complex events without relevant process variables. Since the interactive CPD algorithm always finds the segments containing process variables with similar behavior/characteristics/shape, this will work for the simple events with relevant process variables, but this will not work for the complex events without relevant process variables. However, it could be possible that the complex event may correspond to different sets of relevant process variables in different batches, and therefore, it's worthwhile investigating that if repeating the interactive CPD algorithm multiple times for the same complex event by identifying different segments of relevant process variables each time for the complete dataset will work or not.
- In the experiments with the interactive CPD using the VLSW search method and the DTW distance as the cost function, the accuracy decreases in case of events having noisy process variables as the shape of these variables changes due to the noise. The accuracy also decreases when the process variables in an event across different batches have a similar shape but there is a large difference in their amplitudes because DTW distance is sensitive to the shape as well as the amplitude. However, other cost functions in place of DTW distance could be applied like global alignment kernel (GAK) and autoencoder.
- Moreover, in the experiments with the VLSW search method and the DTW distance as the cost function, the results in the multivariate case are more accurate than that of the univariate case except for the case in which the event corresponds to the noisy process variables. The accuracy of the results in the multivariate case further decreases as compared to the univariate case in the event with noisy process variables because the multivariate DTW distance increases due to the multiple noisy process variables.
- The computation time was more for the cost function with the DTW distance than that of the prediction error. The time for computing was also more for the longer duration events than that for the shorter duration events. Moreover, the computation time increases with the increase in the length of the user-selected segment as well as with the increase in the minimum and the maximum deviation parameters, but it decreases with the increase in the steps parameters. The decrease in the steps parameter increases the accuracy of the results but also increases the computation time.

5.2 Future Works

In this section, different future works with respect to the further evaluation, the algorithm, the prototype, and the extension to other use cases are discussed.

5.2.1 Further Evaluation

In this thesis, the experiments with the interactive CPD algorithm for the events identification using the VLSW search method and the DTW distance as the cost function are performed, evaluated, and discussed on two different datasets: the simulated dataset and the real-life dataset. Therefore, as future work, it will be interesting to see how this algorithm will perform on some other simulated or real-life time series datasets containing process data.

5.2.2 Algorithm

From the algorithm point of view, following could be done as future work:

- For events identification with interactive CPD approach using the PELT search method, to improve the result from the first and the last batch, zero padding at the start and at the end of the dataset could be applied.
- For events identification with interactive CPD approach using the VLSW search method and the DTW distance as the cost function, other methods could be used for setting the cost threshold and separating the sliding window segments with lower DTW distances into multiple groups which are far from one another.
- In the interactive CPD approach, currently only one user-selected segment is incorporated into the algorithm for events identification. For future works, the provision for incorporating multiple such user-selected segments or a reference segment created from these user-selected segments into the algorithm could be implemented.
- For the identification of the complex events without relevant process variables, the interactive CPD algorithm could be repeated multiple times for the same complex event by identifying different segments of relevant process variables each time for the complete dataset because it could be possible that the complex event may correspond to different sets of relevant process variables in different batches.
- Along with the VLSW search method, other similarity metrics like the global alignment kernel (GAK) or the error from the embedding learning models such as autoencoders could be used as the cost function in place of the DTW distance. Some other techniques or approaches such as classification models like support vector machines (SVM) classifier, K nearest neighbors (KNN) classifier, and decision trees, could also be applied in the interactive CPD algorithm. These approaches may address the problem of decreased accuracy when DTW distance is applied to noisy process data or when the trajectories of process variables of a given event have a large difference in their amplitudes across different batches.

5.2.3 Prototype

In the UI prototype, following future works could be performed:

- To facilitate parallel execution of the prototype so that different users can work with the prototype at the same time without interfering with one another, a session management facility for the users could be implemented.
- Facility to provide visualization for multiple generated events on the same plot could be implemented.
- For the identification of the complex events without relevant process variables, functionality to generate artificial labels for the single complex event by running the complete process flow multiple times while avoiding the generation of overlapped labels could be implemented.
- Facility to incorporate multiple user-selected segments for the identification of a single event could be implemented.
- Functionality to use multiple datasets at the same time could be implemented.
- Facility to identify multiple events at the same time could also be implemented in the prototype.
- Other functionalities required by the future works from the algorithm point of view could be implemented.

5.2.4 Extension to other use cases

In this thesis, the interactive CPD approach is applied for the identification of events in the process data. But as future work, this approach can also be applied to other use cases such as searching for anomalies or other potential changes like cleaning, maintenance cycle, etc. in the process data. It would be interesting to see the results of this approach in such use cases.

6 Conclusion

The study in this thesis has addressed the problem of the identification of the events for time series process data by using the interactive change point detection approach. The operating conditions in a process industry can change from time to time. So, it becomes important to keep track of these changing conditions or events. Additionally, for the analysis of the batch process data, the identification of events is important. But due to the inaccurate or missing batch events data from the batch process, it is difficult to use a supervised algorithm for the events identification. One approach for identifying the batch events is through unsupervised change point detection but the results in this case largely depend on the choice of search methods, cost functions, and penalty values. The research in this thesis found out that despite tuning the different parameters, the desired results were not achieved due to the dynamic, non-linear, and non-stationary nature of the process variables. Moreover, the accuracy of the results was decreased further for the multiple batches of data. Therefore, using the unsupervised approach for change point detection is not sufficient. So, to overcome the drawbacks of unsupervised CPD, a new semi-supervised approach for change point detection is introduced in this thesis, i.e., interactive change point detection. In this approach, user feedback is taken into account for generating both initial and updated results. Since the ground truth events may or may not be available, therefore, the interactive CPD approach depends on the feedback from the process experts. The feedback from the process experts is based on their knowledge of the events and the characteristics of the process variables. This useful user feedback is incorporated into the interactive change point detection algorithm to generate artificial labels for the events. Furthermore, different tuning parameters are used in this interactive CPD approach to update the results through user feedback and increase the accuracy of the generated artificial labels for the events. Additionally, a new search method is introduced and implemented in this thesis, i.e., the varying length sliding window (VLSW) search method. The interactive CPD approach is using this new VLSW search method and the DTW distance as the cost function for the events identification problem. Along with this, a UI prototype is also developed in this thesis for the complete interactive process.

The proposed algorithm was applied to two datasets: the simulated dataset and the real-life dataset. The experiments in this thesis with these two datasets for the identification of the events show that the interactive CPD results are more accurate than the unsupervised CPD results due to the introduction of user feedback in the interactive CPD approach. Moreover, this interactive change point detection algorithm and the developed prototype has the potential for extensions to a variety of other use cases as well.

7 Abbreviations

C

CPD	Change Point Detection

D

DTW	Dynamic Time Warping

P

PELT	Pruned Exact Linear Time

V

VLSW	Varying Length Sliding Window

8 References

[1] R. Gedda, L. Beilina and R. Tan, "Change point detection for process data analytics," *arXiv,* 2021.

[2] J. Mancuso, "Ecolink," 21 Aug 2015. [Online]. Available: https://ecolink.com/info/chemical-manufacturing-batch-processing-vs-continuous-processing/. [Accessed 11 Feb 2022].

[3] C. Truong, L. Oudre and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing,* 2019.

[4] R. Rendall, L. H. Chiang and M. S. Reis, "Data-driven methods for batch data analysis – A critical overview and," *Computers and Chemical Engineering,* 2019.

[5] L. L. Zheng, T. J. M. Avoy, Y. Huang and G. Chen, "Application of Multivariate Statistical Analysis in Batch Processes," *Industrial & Engineering Chemistry Research,* 2001.

[6] Schlags, D. Neogi and C. E., "Multivariate Statistical Analysis of an Emulsion Batch Process," *Industrial & Engineering Chemistry Research,* 1998.

[7] L. H. Chiang, R. Leardi, R. J. Pell and M. B. Seasholtz, "Industrial experiences with multivariate statistical analysis of batch process data," *Chemometrics and Intelligent Laboratory Systems,* 2006.

[8] S.-P. Reinikainen and A. Höskuldsson, "Multivariate statistical analysis of a multi-step industrial processes," *Analytica Chimica Acta,* 2007.

[9] M. Ramos, J. Ascencio, M. V. Hinojosa, F. Vera, O. Ruiz, M. I. Jimenez-Feijoó and P. Galindob, "Multivariate statistical process control methods for batch production: a review focused on applications," *Production & Manufacturing Research,* 2021.

[10] O. Marjanovic, B. Lennox, D. Sandoz, K. Smith and M. Crofts, "Real-time monitoring of an industrial batch process," *Computers and Chemical Engineering,* 2006.

[11] Y. Yao and F. Gao, "Phase and transition based batch process modeling and online monitoring," *Journal of Process Control,* 2008.

[12] N. Niang, F. S. Fogliatto and G. Saporta, "Batch Process Monitoring by Three-way Data Analysis Approach," 2009.

[13] Z. Ge, F. Gao and Z. Song, "Batch process monitoring based on support vector data description method," *Journal of Process Control,* 2011.

[14] Z. Ge and Z. Son, "Bagging support vector data description model for batch process monitoring," *Journal of Process Control,* 2013.

[15] M. Yao, H. Wang and W. Xu, "Batch process monitoring based on functional data analysis and support vector data description," *Journal of Process Control,* 2014.

[16] Q. Jiang, F. Gao, H. Yi and X. Yan, "Multivariate Statistical Monitoring of Key Operation Units of Batch Processes Based on Time-Slice CCA," *IEEE Transactions on Control Systems Technology,* vol. 27, no. 3, pp. 1368 - 1375, 2019.

[17] J. Yu, "Multiway discrete hidden Markov model-based approach for dynamic batch process monitoring and fault classification," *AIChE Journal,* vol. 58, no. 9, pp. 2714-2725, 2012.

[18] J. González-Martínez, J. Westerhuis and A. Ferrer, "Using warping information for batch process monitoring and fault classification," *Chemometrics and Intelligent Laboratory Systems,* vol. 127, pp. 210-217, 2013.

[19] H. Rostami, J. Blue and C. Yugma, "Automatic equipment fault fingerprint extraction for the fault diagnostic on the batch process data," *Applied Soft Computing,* vol. 68, pp. 972-989, 2018.

[20] M. Onel, C. A. Kieslich, Y. A. Guzman, C. A. Floudas and E. N. Pistikopoulos, "Big data approach to batch process monitoring: Simultaneous fault detection and diagnosis using nonlinear support vector machine-based feature selection," *Computers & Chemical Engineering,* vol. 115, pp. 46-63, 2018.

[21] C. Peng, R. W. Lu, O. Kang and W. Kai, "Batch process fault detection for multi-stage broad learning system," *Neural Networks,* vol. 129, pp. 298-312, 2020.

[22] L. Zhao and X. Huang, "Slow Time-Varying Batch Process Quality Prediction Based on Batch Augmentation Analysis," *New Frontiers in Industry 4.0,* 2022.

[23] W. Sun, Y. Meng, A. Palazoglu, J. Zhao, H. Zhang and J. Zhang, "A method for multiphase batch process monitoring based on auto phase identification," *Journal of Process Control,* 2011.

[24] S. K. Maiti, R. K. Srivastava, M. Bhushan and P. P. Wangikar, "Real time phase detection based online monitoring of batch fermentation processes," *Process Biochemistry,* vol. 44, no. 8, pp. 799-811, 2009.

[25] F. A. P. Peres, T. N. Peres, F. S. Fogliatto and M. J. Anzanello, "Strategies for synchronizing chocolate conching batch process data using dynamic time warping," *Journal of Food Science and Technology,* p. 122–133, 2020.

[26] M. Zhang, R. Wang, Z. Cai and W. Cheng, "Phase partition and identification based on kernel entropy component analysis and multi-class support vector machines-fireworks algorithm for multi-phase batch process fault diagnosis," *Transactions of the Institute of Measurement and Control,* vol. 42, no. 12, pp. 2324-2337, 2020.

[27] E. Quatrini, F. Costantino, G. D. Gravio and R. Patriarca, "Machine learning for anomaly detection and process phase classification to improve safety and maintenance activities," *Journal of Manufacturing Systems,* vol. 56, pp. 117-132, 2020.

[28] M. L. C. André, B. Satyajeet, P. Kristin and V. I. J. F. M., "Manifold Learning and Clustering for Automated Phase Identification and Alignment in Data Driven Modeling of Batch Processes," *Frontiers in Chemical Engineering,* vol. 2, 2020.

[29] E. S. Page, "Continuous Inspection Schemes," *Biometrika,* vol. 41, no. 1/2, pp. 100-115, 1954.

[30] E. S. Page, "A test for a change in a parameter occurring at an unknown point," *Biometrika,* vol. 42, no. 3-4, p. 523–527, 1955.

[31] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and Information Systems,* p. 339–367, 2017.

[32] S. Liu, M. Yamada, N. Collier and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation," *Neural Networks,* vol. 43, pp. 72-83, 2013.

[33] G. J. v. d. Burg and C. K. Williams, "An Evaluation of Change Point Detection Algorithms," *arXiv,* 2020.

[34] M. D. S. L. C. Vicente, "Model for Batch Process Data Generation - A Benchmark," 2021.

[35] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar and E. Woods, "Tslearn, A Machine Learning Toolkit for Time Series Data," *Journal of Machine Learning Research,* vol. 21, no. 118, pp. 1-6, 2020.

[36] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 26, no. 1, pp. 43-49, 1978.

9 Appendix

Appendix A: Results of Experiment U1

There are 12 experiments in Experiment U1. The results of all these 12 experiments are shown in the following figures. These figures illustrate multiple process variables, events, and detected change points for a single batch from the simulated dataset using unsupervised CPD. The colored traces represent the process variables, colored areas represent the events and vertical dashed lines represent the detected change points.

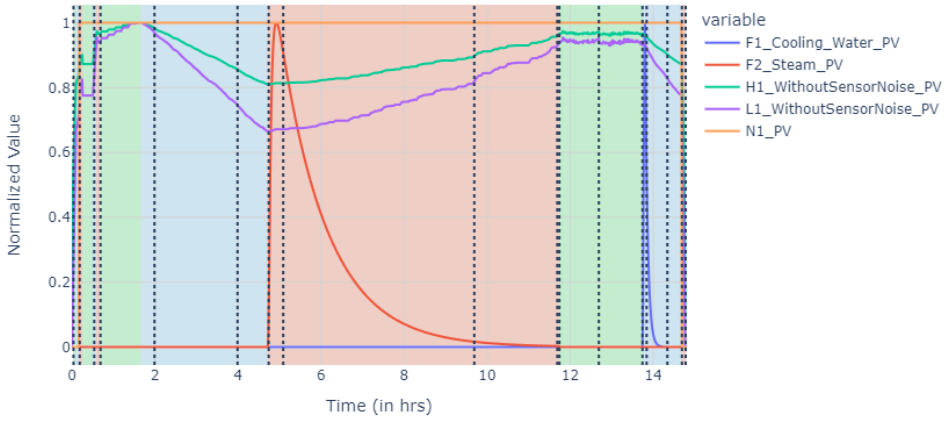

Figure 9.1 Process variables, events, and detected change points obtained from Experiment U1.1

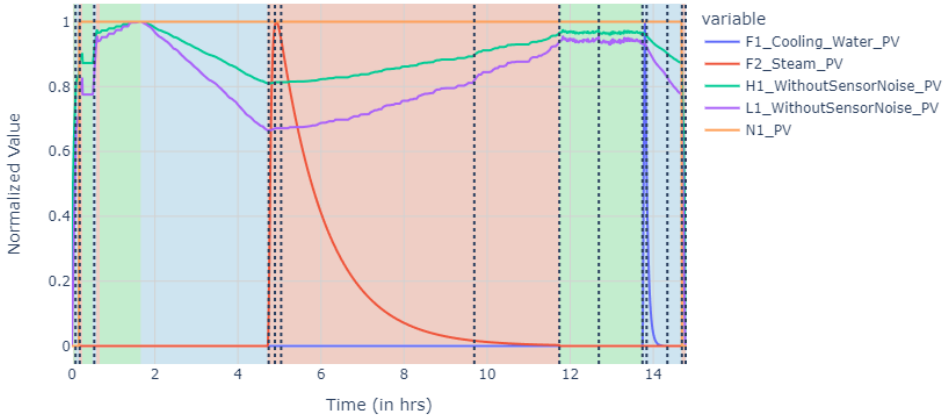

Figure 9.2 Process variables, events, and detected change points obtained from Experiment U1.2

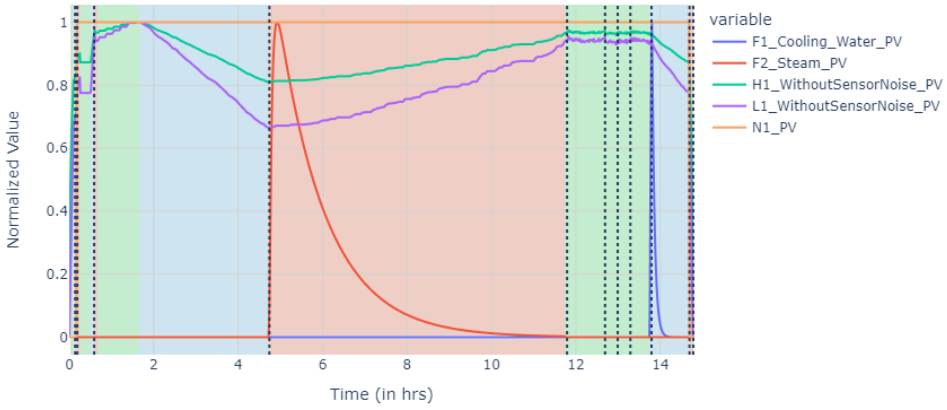

Figure 9.3 Process variables, events, and detected change points obtained from Experiment U1.3

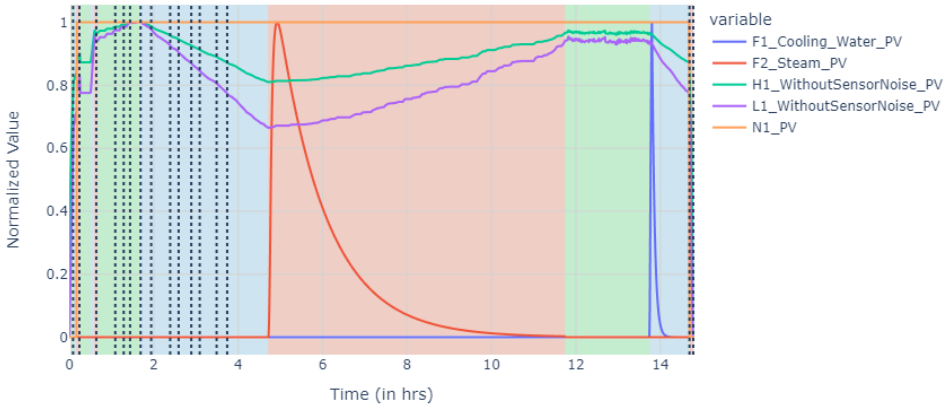

Figure 9.4 Process variables, events, and detected change points obtained from Experiment U1.4

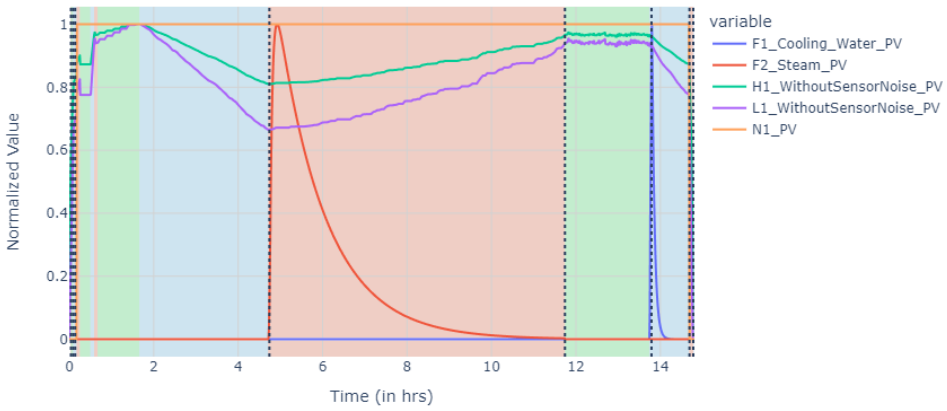

Figure 9.5 Process variables, events, and detected change points obtained from Experiment U1.5

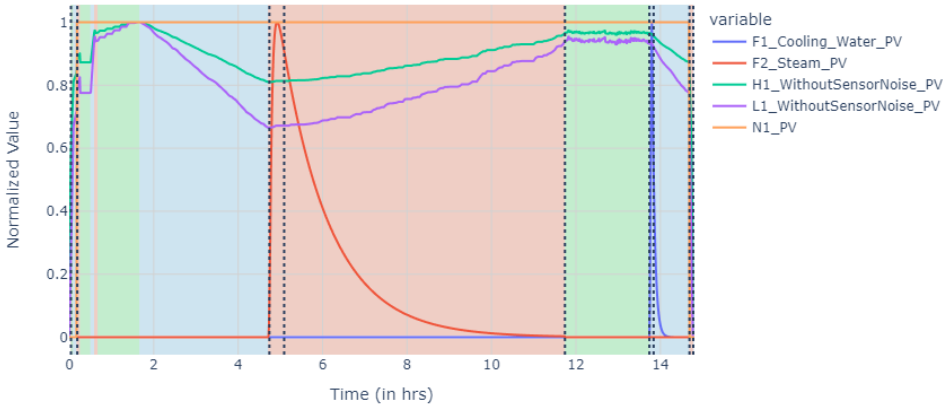

Figure 9.6 Process variables, events, and detected change points obtained from Experiment U1.6

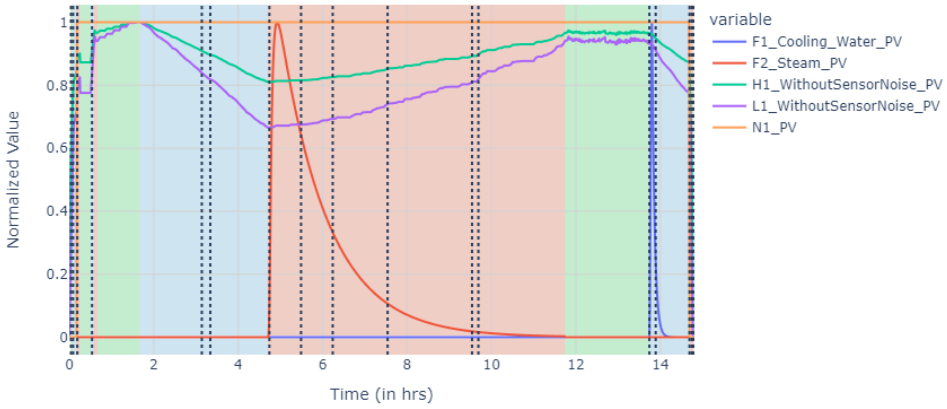

Figure 9.7 Process variables, events, and detected change points obtained from Experiment U1.7

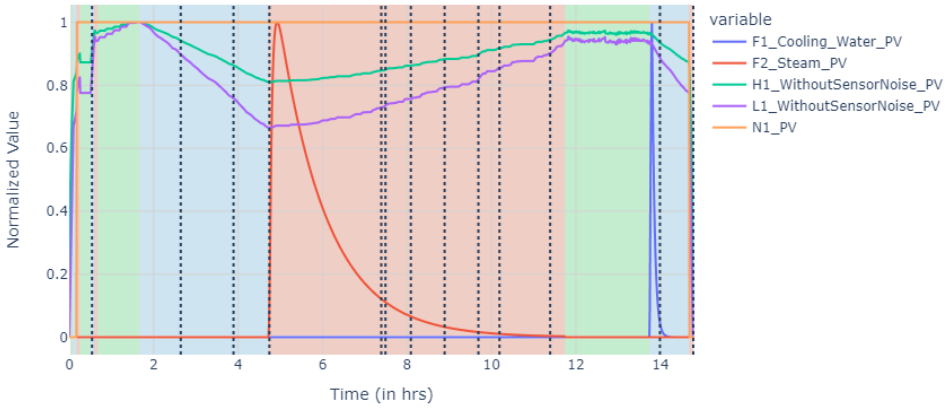

Figure 9.8 Process variables, events, and detected change points obtained from Experiment U1.8

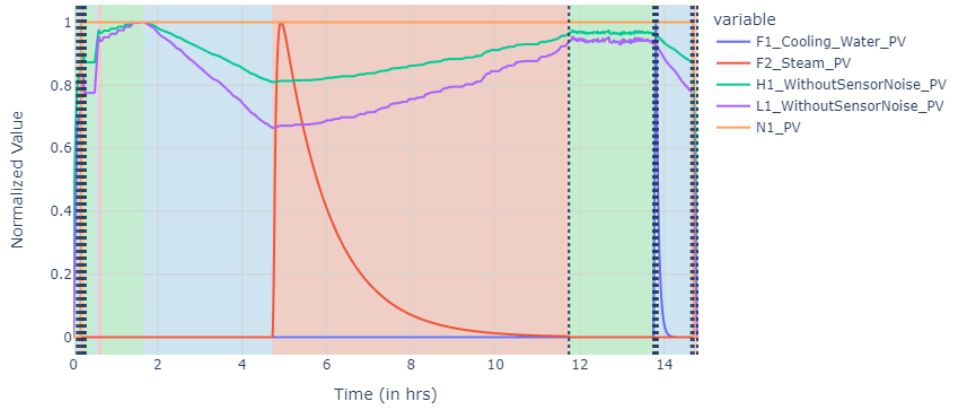

Figure 9.9 Process variables, events, and detected change points obtained from Experiment U1.9

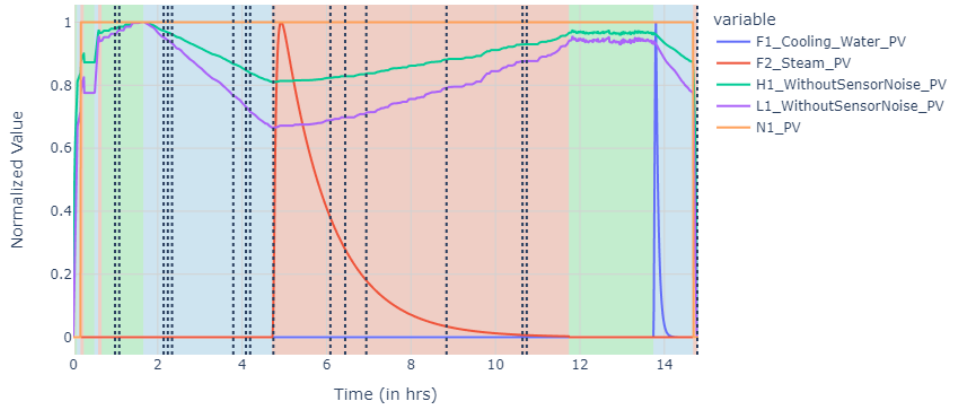

Figure 9.10 Process variables, events, and detected change points obtained from Experiment U1.10

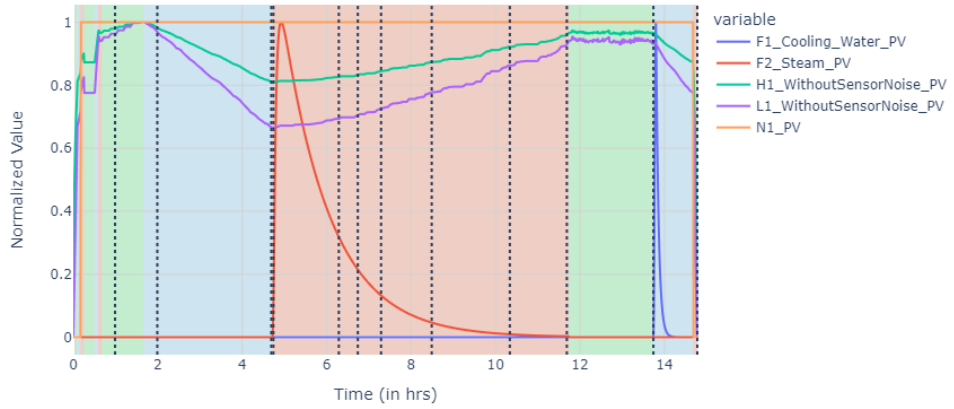

Figure 9.11 Process variables, events, and detected change points obtained from Experiment U1.11

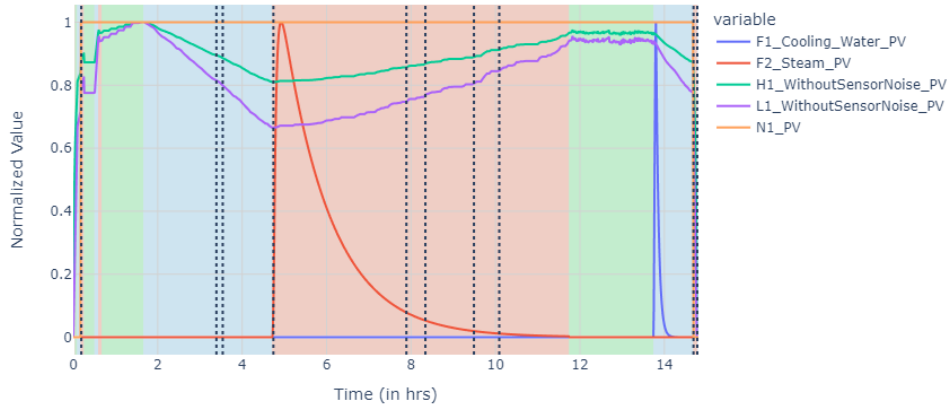

Figure 9.12 Process variables, events, and detected change points obtained from Experiment U1.12

Appendix B: Results of Experiment U2

There are 5 experiments in Experiment U2. The results of all these 5 experiments are shown in the following figures. These figures illustrate multiple process variables, events, and detected change points over a range of penalties for a single batch from the simulated dataset using unsupervised CPD. The colored traces represent the process variables, the colored areas represent the events, and the vertical solid lines represent the detected change points. The height of the lines indicates the range of penalty values in which that detected change point exists.

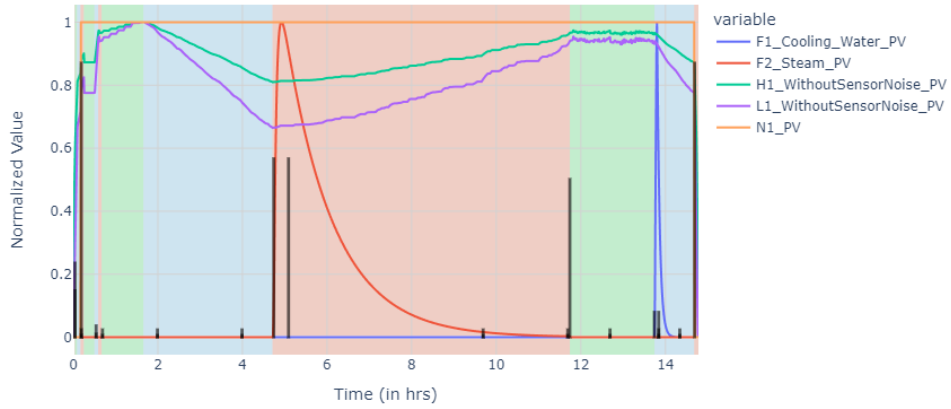

Figure 9.13 Process variables, events, and detected change points obtained from Experiment U2.1

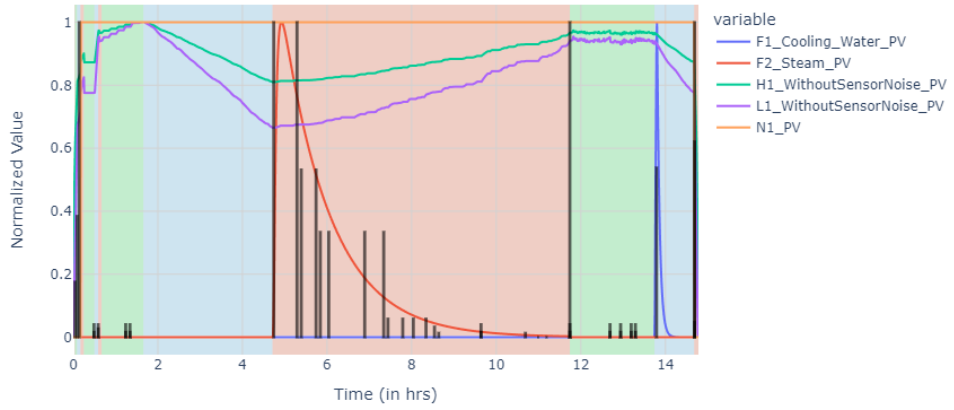

Figure 9.14 Process variables, events, and detected change points obtained from Experiment U2.2

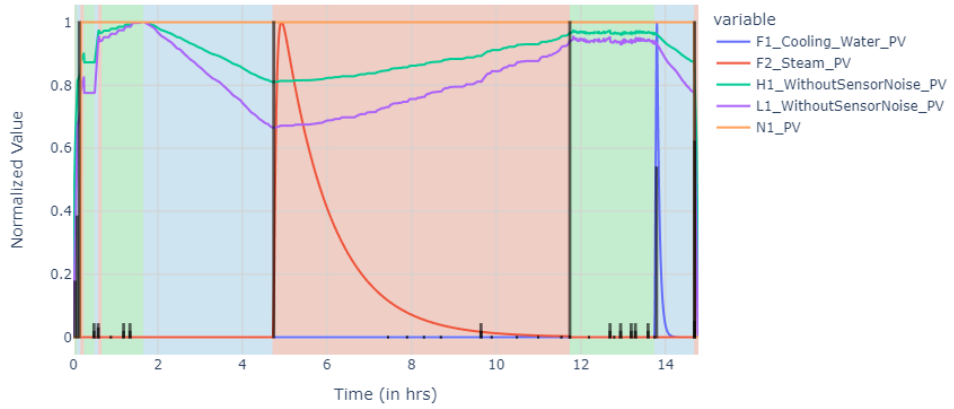

Figure 9.15 Process variables, events, and detected change points obtained from Experiment U2.3

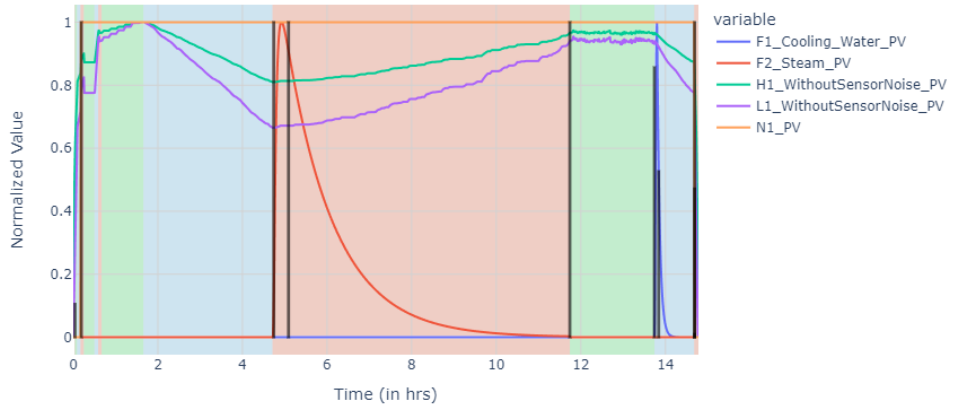

Figure 9.16 Process variables, events, and detected change points obtained from Experiment U2.4

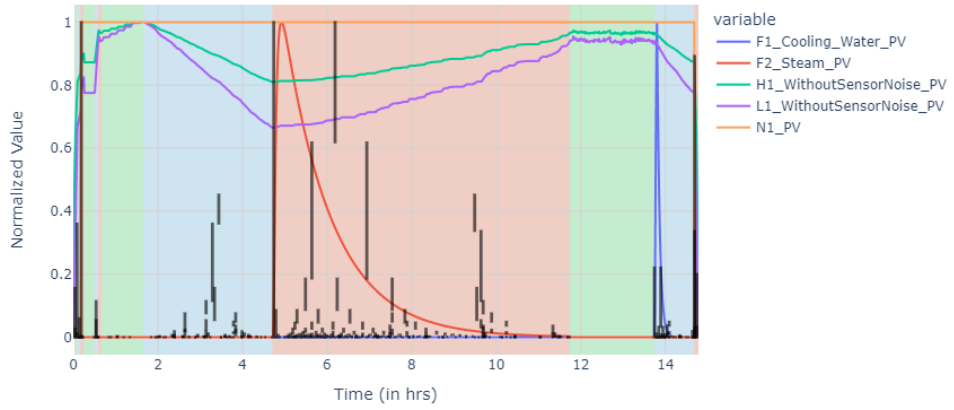

Figure 9.17 Process variables, events, and detected change points obtained from Experiment U2.5

Appendix C: Results of Experiment U3

There are 4 experiments in Experiment U3. The results of all these 4 experiments are shown in the following figures. These figures illustrate multiple process variables, events, and detected change points over a range of penalties for multiple batches from the simulated dataset using unsupervised CPD. The colored traces represent the process variables, the colored areas represent the events, and the vertical solid lines represent the detected change points. The height of the lines indicates the range of penalty values in which that detected change point exists.

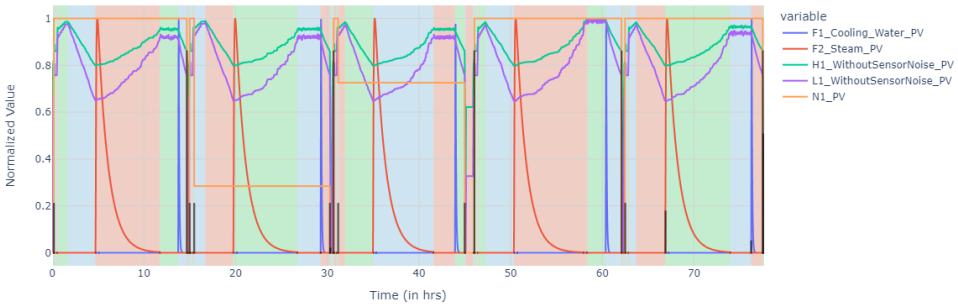

Figure 9.18 Process variables, events, and detected change points obtained from Experiment U3.1

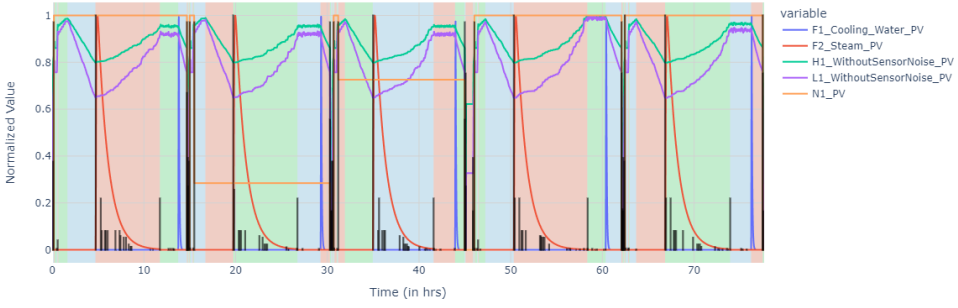

Figure 9.19 Process variables, events, and detected change points obtained from Experiment U3.2

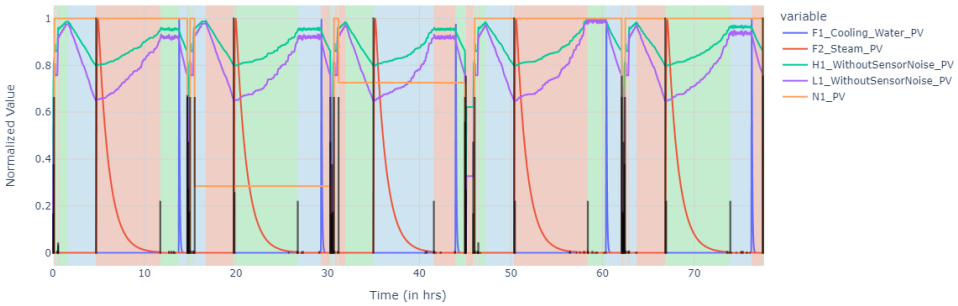

Figure 9.20 Process variables, events, and detected change points obtained from Experiment U3.3

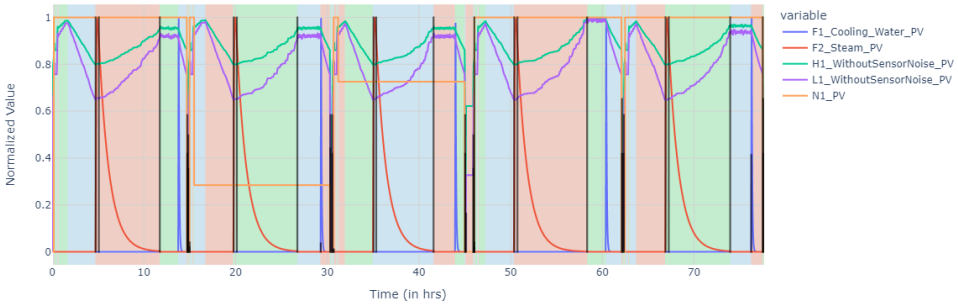

Figure 9.21 Process variables, events, and detected change points obtained from Experiment U3.4