International Journal of Agricultural and Environmental Information Systems
Volume 11 • Issue 1 • January-March 2020

Concept Drift Detection in Data Stream Clustering and its Application on Weather Data

Namitha K., Artificial Intelligence and Computer Vision Lab, Department of Computer Science, Cochin University of Science and Technology, Kochi, Kerala, India

Santhosh Kumar G., Artificial Intelligence and Computer Vision Lab, Department of Computer Science, Cochin University of Science and Technology, Kochi, Kerala, India

ABSTRACT

This article presents a stream mining framework to cluster the data stream and monitor its evolution. Even though concept drift is expected to be present in data streams, explicit drift detection is rarely done in stream clustering algorithms. The proposed framework is capable of explicit concept drift detection and cluster evolution analysis. Concept drift is caused by the changes in data distribution over time. Relationship between concept drift and the occurrence of physical events has been studied by applying the framework on the weather data stream. Experiments led to the conclusion that the concept drift accompanied by a change in the number of clusters indicates a significant weather event. This kind of online monitoring and its results can be utilized in weather forecasting systems in various ways. Weather data streams produced by automatic weather stations (AWS) are used to conduct this study.

KEYWORDS

Clustering, Concept Drift, Data Streams, Short-Range Weather Forecasting

INTRODUCTION

With the advancement in hardware and software technology, the number of applications producing large volume data streams is ever increasing. These data streams become useful to the industry and society only when the valuable information contained in them is extracted. Various data stream mining algorithms are available for this purpose. Capability to process the data in a single scan, requirement of online and incremental updation of the models, adaptation to the concept changes etc. are some challenges while designing learning algorithms for data streams. Clustering is an important machine learning task applied on data streams to gain useful insights into the natural groupings in data. It also helps to track the development of various phenomena in application fields like meteorology, healthcare and astrophysics (Bhatnagar, 2009).

The main advantage of data stream mining is that it assumes the data source or the process generating the stream is not stationary. According to the changes in the environment that produces

DOI: 10.4018/IJAEIS.2020010104

the stream, the underlying data distributions change over time. Consequently, these changes might affect the inter relationship between input and output variables (Gama, 2014) leading to 'Concept Drift' (Widmer and Kubat, 1996; Gama, 2014). In these situations, the learned model becomes obsolete and the prediction accuracy reduces considerably. Weather data is a classic example where the concept can change over time (Widmer and Kubat, 1996). Depending on the seasons, weather prediction rules might exhibit a change, i.e., the same relation might not hold well in all the seasons.

This paper proposes a framework for the online clustering of data streams. It performs concept drift detection and cluster evolution monitoring to generate a warning on the dynamic changes taking place in the environment of the stream. Studies revealed that concept drift accompanied by clustering structure changes often imply important physical events. Recording such inter-relationships between the occurrence of concept drift and the corresponding physical events over a period of time can help the prediction of these physical phenomena by cluster analysis.

The proposed framework has components to cluster the stream online, detect concept changes and track the evolution of clusters. Before performing the clustering, the best value for the 'number of clusters', k is computed dynamically. This is done with the intention to identify the changes in the clustering structure for the recently arrived data. As the source of the data is highly dynamic, the clustering structure also might exhibit a corresponding change and fixing the value of k limits the ability to capture such changes in the clustering structure.

The utility of this framework is studied by applying it on weather data. Concept drifts, changes in the best value of k and the cluster evolutions are being monitored to understand their relationship with the physical weather phenomena. Nowadays most of the weather monitoring equipment produces high-speed data with a large number of variables. Weather stream produced by the Automatic Weather Station (AWS) at the Advanced Centre for Atmospheric Radar Research (ACARR) of Cochin University of Science and Technology, Kerala, India is used for this study. Data is collected at one-minute intervals and it contains the weather parameters like temperature, relative humidity, wind speed, wind direction, radiation, pressure, rainfall, etc. Monsoons are the most important weather phenomenon as far as the Indian region is concerned. Hence the framework is used to study the interrelationship between the changes in the clustering structure and the evolution of the south-west monsoon.

The paper is structured as follows. Section 2 refers to the background of this work. Section 3 details the proposed framework with subsections on each component of the framework. Section 4 explains the application of the proposed framework on weather data. Section 5 details about the experiments and results followed by 'discussion and future work' in section 6. The paper is concluded in section 7.

BACKGROUND

Detecting Concept Drift in Data Stream Clustering

Concept drift detection and adaptation are studied more in the context of supervised learning. As (Gama, 2014) states in his survey, the problem of concept drift handling has a much wider scope and it is applicable to clustering problems as well. Research in this direction is still in the starting phase. Surveys conducted on data stream clustering (Ghesmoune et al., 2016; Silva et al., 2013) also point to the fact that explicit concept drift detection and adaptation are rarely done in data stream clustering algorithms.

In supervised learning problems, concept drift can be defined as a change in the joint probability distribution P(X, Y), where X denotes a random variable over vectors of attribute values and Y denotes a random variable over class labels (Webb, 2016). But in the case of unsupervised learning, since the instances are not labelled, the definition of concept drift is modified as a change in the probability distribution P(X). Hence statistical methods of change detection are usually used in data

stream clustering problems to identify concept drift. Silva et al. (2017) and Sakamoto et al. (2016) are the recent works that discuss the problem of concept drift detection in data stream clustering. Sakamoto et al. (2016) propose a combination of Drift Detection Method (DDM) and Page-Hinkley Test (PHT) to handle drift whereas Silva et al. (2017) use Page Hinkley Test alone. As discussed in (Chen and Liu, 2006), changes in the clustering structures can give a clue on new emerging patterns and thus help in the prediction of real events.

Cluster Evolution Analysis

Cluster evolution analysis done as part of this work is inspired by the cluster transition monitoring models available in the literature. Cluster transitions happening over a period of time are monitored and their relationship to the physical events has been analysed. MONIC (Spiliopoulou et al., 2006) and MEC (Oliveira & Gama, 2010) are the basic frameworks available for cluster transition modelling. MONIC monitors the overlap between clusters in the consecutive time points and detects external cluster transitions based on this. MONIC is later extended as MONIC+ (Ntoutsi et al., 2009) with the definition of different types of clusters. It re-defines the overlap and transition of clusters based on this.

Machine Learning Applied to Weather Solutions

Weather monitoring systems produce unbounded streams of data, at a large scale and at a rapid rate. Companies offering weather solutions are using stream processing engines like Apache Storm (http://storm.apache.org/) for processing such streams at real-time. Though stream processing platforms are well-known to process weather data, stream mining solutions are not extensively tried for weather forecasting problems.

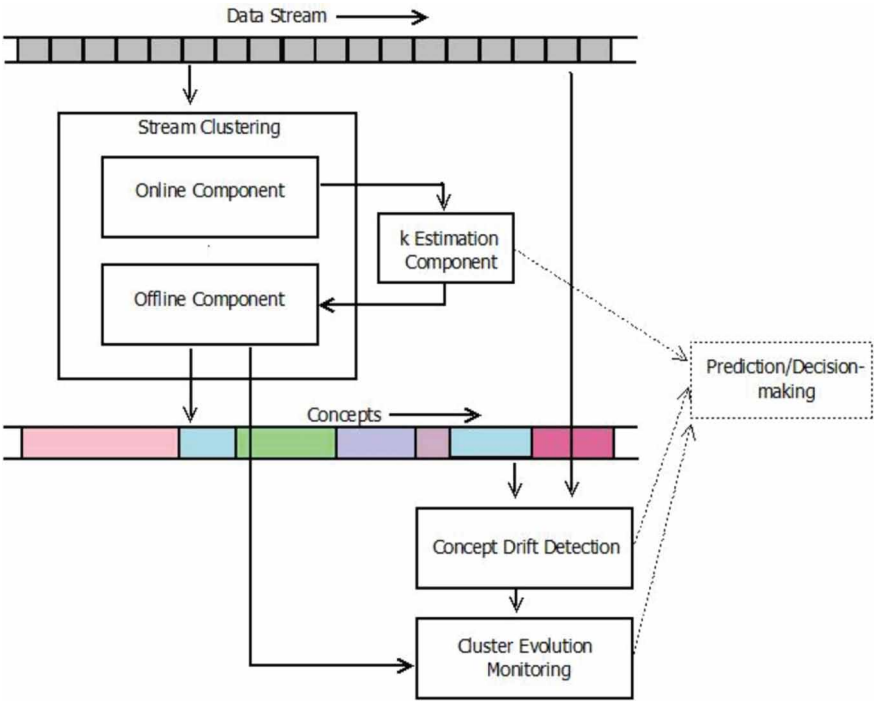
However, machine learning methods are applied to weather forecasting problems to a great extent. Two most important machine learning methods tried for weather prediction problems are artificial neural networks (ANN) and support vector machines (SVM). Hung et al. (2008) proposed a method of using an Artificial Neural Network for short-range rainfall forecasting. An earlier work done by Guhathakurta (2006) is using ANN architecture for long-range prediction of annual Monsoon rainfall. Literature includes plenty of papers which tried ANN for solving weather-related problems (Kuligowski & Barros, 1998; Smith et al., 2009; Kenabatho et al., 2013). Radhika and Shashi (2009) illustrate an attempt to make use of Support Vector Machine for temperature prediction. An extensive review on applications of Support Vector Machine in the field of hydrology can be found in (Raghavendra and Deka, 2014). Good account of papers is available in the literature which discusses the applicability of Support Vector Machines on various weather prediction tasks (Liu et al., 2012; Nayak and Ghosh, 2013).

THE PROPOSED FRAMEWORK

Figure 1 shows the proposed framework. Online stream clustering is performed using a two-level clustering algorithm. It has an online streaming component which generates the synopsis of the stream and an offline component that generates the clusters from this synopsis. Computation of the best value of *k* is performed between these two phases to generate that many clusters during the offline clustering.

For stream clustering problems, a concept is defined as the probability distribution P(X) where X is the random variable over vectors of attribute values (Webb, 2016). Clustering or the partition produced at a particular time point is nothing but a representation of the data distribution P(X) at that moment. Clustering can be considered as a probability mixture model with each cluster representing one component of the mixture. Since CluStream uses k-means for offline clustering, the clusters generated are convex in shape and follows a normal distribution. So, the n^{th} component of the mixture model can be written as:

Figure 1. Overview of the proposed framework



$$\varphi_n\left(x \mid \mu_n, \Sigma_n\right) \equiv \frac{\exp\left\{-\frac{1}{2}(x-\mu_n)^T \Sigma_n^{-1}\left(x-\mu_n\right)\right\}}{\sqrt{\det\left(2\pi\Sigma_n\right)}}$$

where x is a d-dimensional column vector representing a data item, μ_n the mean vector and Σ_n the covariance matrix (Fraley, 2002).

Since the clustering depicts the data distribution, the clustering generated by the offline component of the framework is treated as the concept learnt from the data. At a time only one concept is active and the fitness of this concept to the recently arrived data is always being monitored. Concept drift detection module is used for this purpose. Current concept and the samples retrieved from the stream are the input to this module. Whenever the change is detected, a new clustering is generated from the recent data items and the current concept is replaced by this new concept. It has to be noted that clusters undergo various transitions during these concept drifts. Cluster evolution analysis is done to understand the nature of these transitions. Information obtained from the concept drift detection module, k estimation module and the cluster evolution analysis can be combined together to support the prediction process. An overview of these components is given in the next subsections.

Stream Clustering

While incorporating an explicit concept drift detection module to an existing stream clustering algorithm, the major concern is how to blend the drift detection methodology with online learning. Before going to those details, a brief discussion on the stream clustering algorithm used in this framework is given.

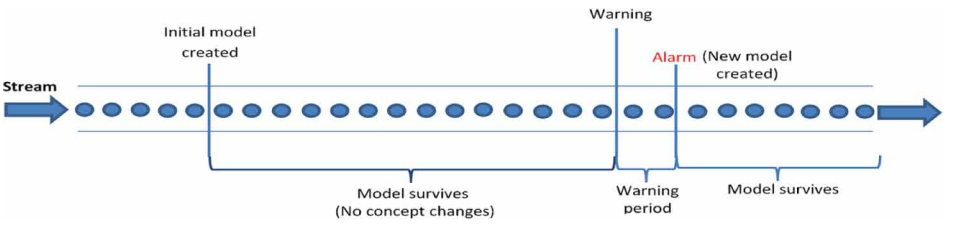
Clustering is a method commonly used for statistical analysis of data. It is defined as the task of grouping a set of data points into clusters such that points within one cluster are similar to each other and points from different clusters are dissimilar (Leskovec and Rajaraman, 2014). In our experiments, stream clustering is done using CluStream (Aggarwal et al., 2003), a distance-based clustering algorithm. CluStream has good clustering accuracy and an in-built capability to deal with evolving data streams. It divides the clustering process into online and offline components, with the online component generating micro-clusters and the offline component generating macro-clusters from micro-clusters. Micro-clustering is the method of storing the summary statistics of the stream. A micro-cluster is stored as a five element tuple - altogether representing the summary of data points in that cluster. This summary serves the purpose of calculating the characterising features of the cluster, like its centre, radius and diameter at any point of time. Another point to be noted is that, summary of the timestamp information of the members is also stored in each micro-cluster. This information will be required for computing the recency of a micro-cluster, when removal of some micro-clusters from the memory is necessary as part of the evolution of the stream. On the arrival of each data record of the stream, it's similarity to the existing micro-clusters is computed. If found similar, it is added to the most similar micro-cluster, otherwise a new micro-cluster is formed to represent this data record, giving an impression that stream is evolving.

The offline phase or so-called macro-clustering phase is the one which creates the real outcome of the stream clustering process. When the time horizon applicable to macro-clustering is provided, it retrieves the micro-clusters relevant to that time period and does an offline clustering of these micro-clusters. During this clustering phase, micro-clusters are treated as single data points represented by their centres. Initial cluster centres or the seeds are not selected randomly, but the more weighted k micro-clusters are chosen as seeds. Further, we have introduced a modification to this algorithm. The value of 'k' in macro-clustering phase is not fixed, it is calculated dynamically based on an algorithm which is detailed in the next session.

In CluStream algorithm, macro-clustering process is initiated upon user request and it generates macro-clusters relevant for the period requested by the user. As demanded by the proposed framework, certain deviations are adopted and they are summarized as follows. Upon creation of a group of macro-clusters, that particular clustering is treated as the active model as long as the underlying data distribution persists, or in other words, until the same concept survives. An explicit concept drift detection methodology based on Page-Hinkley Test (PHT) is adopted here, which monitors the stream continuously to find out the probable concept changes. First a warning is given and if still the stream continues to deviate from the current concept, an alarm is triggered to declare the concept drift. When a concept drift is flagged by this algorithm, next macro-clustering process is initiated. Before running the macro-clustering phase, value of 'k' is estimated dynamically. On creation of the new k macro-clusters, the new model is established.

Overview of the above-mentioned process is shown in Figure 2. The following three processes will be running continuously throughout the stream:

Figure 2. Overview of the concept drift detection process



- Online maintenance of micro-clusters;
- Storing the micro-clusters in memory;
- Page-Hinkley Test to detect concept drift.

And when concept drift detection algorithm flags an alarm, the following four processes are executed:

- Extraction of the micro-clusters relevant to the warning period i.e., the time period between warning flag and alarm flag;
- Computation of the value of k, i.e., the number of macro-clusters;
- Creation of the new model;
- Recalculation of the Page-Hinkley Test (PHT) parameters.

The flexibility provided by the CluStream algorithm to choose the time horizon of macro-clustering, is utilised to get the macro-clustering over the warning period.

Calculating Number of Clusters Online

Data stream clustering algorithms based on the principle of k-means usually work on the assumption that the number of clusters 'k' is provided by the user and it is fixed throughout the stream. The CluStream algorithm also works on this assumption. But it is obvious that streams can undergo unforeseen changes, especially when the process generating the stream is non-stationary. The literature points to a few stream clustering algorithms which considered this issue (Silva and Hruschka, 2011; Faria et al., 2012; Silva et al., 2017). Change in the number of clusters is an indication of a change in data distribution, which in turn implies a possible drift in the prevailing physical conditions. Results supporting this fact can be found in the Experiments section.

To find the best value of k, there should be a method to assess the relative quality of different data partitions (Naldi et al., 2009; Silva et al., 2017). Hence the optimum value of k and successively the data partition with best quality can be achieved. There are different methods for assessing the relative quality of data partitions and the most popular one among them is simplified silhouette. Comparison studies also reveal (Vendramin et al., 2013) that this is the best method for assessing the quality of a data clustering. Silhouette (Rousseeuw, 1987) value of a clustering is computed as follows. Consider x_i is an element of cluster C_a, and $a(x_i)$ is the average dissimilarity of x_i to all other elements of C_a. For each cluster C_b other than C_a, compute the average distance between x_i and the elements in C_b. The cluster having the lowest value for this average distance is called the neighbouring cluster of x_i and let this lowest value be denoted as $b(x_i)$. It is generally observed that a good clustering will have low value for $a(x_i)$ and high value for $b(x_i)$. Silhouette can be measured as:

$$s\left(x_i\right) = \frac{b\left(x_i\right) - a\left(x_i\right)}{\max\left\{a\left(x_i\right), b\left(x_i\right)\right\}}$$

In the event that all data points are assigned to their nearest clusters, the value of $s(x_i)$ will be within the interval [0,1]. Bigger value for $s(x_i)$ implies better clustering. To reduce the processing overhead, calculating distance to all the elements of a cluster is replaced by calculating the distance to its centre. Hence $a(x_i)$ is the distance between x_i and its own cluster's centroid. Similarly, to calculate $b(x_i)$, distances between x_i and all other cluster centroids are taken. This variant of a silhouette is called Simplified Silhouette (*SS*). Average of all the $s(x_i)$ over $i = 1, \ldots, N$ is taken as the *SS* value of that partition:

$$SS = \frac{1}{N}\sum_{i=1}^{N} s\left(x_i\right)$$

Partition with the highest value of *SS* is taken as the best clustering. As mentioned earlier, centroids of the micro-clusters selected for macro-clustering are considered to be the data points for calculating Simplified Silhouette as well.

Concept Drift Detection

CluStream has been built with the capabilities to handle an evolving data stream. But it does not make use of any explicit change detection procedure. Explicit change detection is desirable because it helps to deal with the change in a timely manner and recover quickly from the performance drop. Also, this framework needs to assess the clustering structure changes on the identification of a concept drift, which is possible only if drift is signalled explicitly. Page-Hinkley Test (PHT) is a standard procedure for change detection (Serakiotou, 1987; Mouss et al., 2004; Sakamoto et al., 2016; Page, 2017; Trust, 2017). A basic principle behind change detection methods is to track how well the existing model fits the recently arriving data points. If the recently arriving data points deviate considerably from the current model, it indicates that the model is becoming unfit to represent the current data partition. In other words, the concept is changing and model needs a reasonable update.

Page-Hinkley Test continuously monitors a parameter which can represent the change. In clustering, the average distance between the data records and their closest cluster centres is the parameter being monitored. Silva et al. (2017) and Zhang et al. (2010) suggest Page-Hinkley Test as the method to be used for explicit change detection in streaming environments. Let D_i be the distance variable whose value is being monitored by Page-Hinkley test. D_i is the distance of data record i to its closest cluster centre in the current data partition. A cumulative variable m_T is calculated as below:

$$m_T = \sum_{i=1}^{T}\left(D_i - \overline{D_i} - \delta\right)$$

where:

$$\overline{D_i} = \frac{\sum_{j=1}^{i} D_j}{i}$$

m_T is the sum of deviation between the observed variable D_i and its mean till moment T. δ is the tolerance level. Two threshold values λ_A and λ_W are used to denote the alarm threshold and warning threshold respectively. Obviously, λ_A should be set greater than λ_W. Page-Hinkley Test gives a warning when the difference between m_T and it's minimum $M_T = \text{Min}(m_i, i = 1, \ldots, T)$ becomes greater than the warning threshold λ_W. i.e., when $m_T - M_T > \lambda_W$. Similarly, it triggers an alarm when this value becomes greater than the alarm threshold λ_A. At receiving each data point from the stream, its distance to the closest cluster in the current partition is measured. The above-mentioned variables are also updated accordingly. Crossing the warning threshold definitely marks the considerable deviation of the recently received objects from the existing partition or points to the fact that the existing partition is getting obsolete. In order to confirm this, the forthcoming data records are monitored until they cross the alarm threshold. This time interval is shown as the 'warning period' in Figure 2 and a concept drift is declared immediately as the alarm threshold is crossed.

The process of macro-clustering, which is responsible for creating the new model, is initiated as a concept drift is detected. (Silva et al., 2017) put forward the idea of adding the data points arriving in the warning period to a buffer and use them for creating the new model. But the proposed framework follows a slightly different methodology of just noting the warning period and extracting only those micro-clusters which are relevant to this time interval. Extracted micro-clusters are then fed as input to the macro-clustering phase. The whole set of PHT parameters, including the threshold values are recalculated after creating the new model.

Cluster Evolution Analysis

Besides the changes in the number of clusters, substantial cluster transitions also occur during concept drifts. Cluster transitions include the disappearance of long-lived clusters, creation of new clusters, merging and splitting of existing clusters and so on. Cluster evolution analysis is performed as part of the experiments, to uncover such cluster level transformations occurring at concept change points. The literature proposes a framework named MONIC, to monitor the cluster transitions (Spiliopoulou et al., 2006). It basically compares the clusters in two consecutive clusterings and finds out the differences and similarities between them, in terms of their internal and external transitions. As part of this study, experiments are conducted particularly to unveil the external transitions, namely - absorption, survival, split, disappearance and creation of a new cluster.

APPLICATION ON WEATHER DATA

Accurate weather forecasting is a basic need of any modern society. One of the greatest impacts of weather forecasting is with the agricultural sector (Sivakumar, 2006; Meza et al., 2008; Wang and Cai, 2009). Unfortunately, the atmospheric system is complex in nature and prediction of weather especially in tropical region is challenging to a great extent.

The two different approaches in objective weather forecasting are numerical weather prediction and statistical weather prediction (Glahn, 1982). The literature points to the good and appreciable results achieved through numerical methods. However, statistical methods cannot be neglected and they are equally powerful in many of the scenarios (Glahn, 1982; Wilks, 2006). Little et al. (2009) mention the competitiveness of statistical methods compared to numerical methods in shorter (a few hours ahead) and longer-term (more than 10 days ahead) weather forecasts. Sometimes numerical and statistical methods are combined to get better results (Krasnopolsky and Fox-Rabinovitz, 2005). Since machine learning is considered as a blend of statistics and artificial intelligence, it is not surprising that machine learning algorithms can also contribute to weather forecasting (Guhathakurta, 2006; Hung et al., 2008; Radhika and Shashi, 2009; Chakraborty et al., 2012; Raghavendra and Deka, 2014). India Meteorological Department (IMD) has developed various statistical models for the long-range prediction of monsoon (Rajeevan et al., 2004; Guhathakurta, 2006). These models forecast for larger regions, either the country as a whole or divide it into three or four broader areas. As mentioned in (Rajeevan et al., 2004), site-specific and short-range forecasting are equally difficult using statistical models. But, short-range forecasting for a smaller locality is very important, particularly when rainfall is considered. There are industries like aviation, which can take advantage of this kind of warnings.

In the experiments, the proposed framework is used to constantly monitor the statistical distribution of weather data at a particular location. Changes in the data distribution will be reflected in the number and distribution of the clusters (Chen and Liu, 2006; Silva et al., 2017) which in turn indicates a change in the underlying weather system. Statistical distribution changes are identified by using a concept drift detection technique in the framework. Thus, the continuous monitoring of the data partition and timely detection of concept drifts help to give hint on sudden changes in the prevailing weather conditions. It can also help the weather forecasting process in many ways. For example, sometimes the concept drift can be a real drift (Gama, 2014), in the sense that it affects

the inter relationship between the predictors and the output variable. This indicates that the existing weather prediction model might need an update to ensure the correctness of its prediction thereafter. Similarly, a concept change point can be used as a landmark to re-analyse the correlation between the available input variables and the output variable.

EXPERIMENTS AND RESULTS

General Experimental Set-Up and Parameter Settings

Experiments are conducted to study the clustering behaviour of weather data and its relationship to the physical phenomena occurring in the environment. Onset and withdrawal of south-west Monsoon and thunderstorms occurrences are the weather phenomena selected for the study. In Cochin, Kerala, normally the onset of southwest monsoon happens during June first week and it withdraws during the beginning of October. Also, unpredictable rain and thunderstorm are common during the summer season. Data collected in the years 2016 and 2017 are used for this study. The Min-Max procedure is performed to normalize the data values to the range [0,1]. The first model will be created in an offline way using the initial 5000 samples of the stream. K-means clustering is the algorithm used in this stage. Thereafter, the model is updated online, along with the stream. Online maintenance of micro-clusters in CluStream algorithm makes sure that the total number of micro-clusters in main memory does not exceed a predetermined limit at any point of time. In these experiments, this limit is fixed to 200.

Automatic Weather Station in ACARR collects most of the weather parameters at three different height levels: 2 meters, 20 meters, and 30 meters. Hence, there is a redundancy in information as far as prediction is concerned. All these parameters might not be relevant for prediction. A dimensionality reduction technique was applied in the beginning to find out the most relevant dimensions or parameters. Temperature, cloud radiation, solar radiation, net radiation, wind speed and wind direction are found related to the rain. Similarly, the temperature, humidity and wind speed are considered for thunderstorm related studies.

Onset of South-West Monsoon

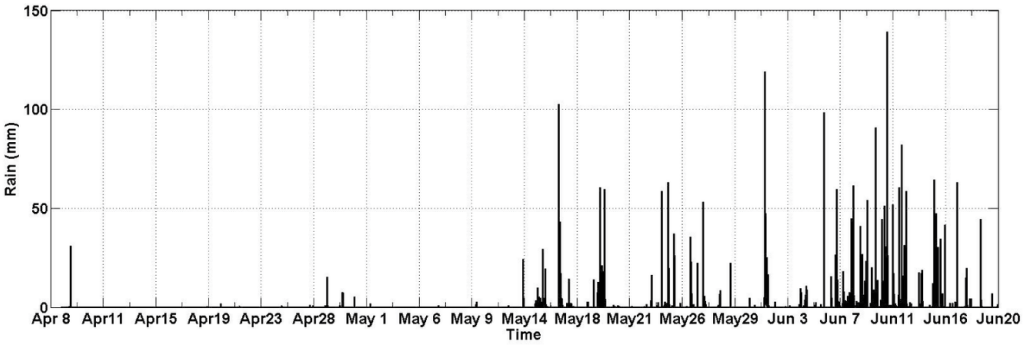
Figure 3(a) shows the rainfall received during the onset of south-west monsoon. As it is evident from this figure, there is a pre-monsoon shower starting from May 14th onwards. Temperature, Solar radiation and most of the weather parameters show a significant difference in distribution during this period. Figure 3(b) shows the concept drift points identified while processing the stream. The number of clusters – k, computed at each concept drift point is also displayed in this figure. Concept drift points are shown overlaid on the plot of temperature to emphasise that change in data distribution is captured by the system.

A change in all the weather parameters can be observed in the week of 16th to 20th May. During the same time, Pre-Monsoon shower started and it prolonged till the monsoon onset during June first week. Figure 3(b) shows that the framework captured these changes and the number of clusters changed suddenly from 4 to 2 during this time.

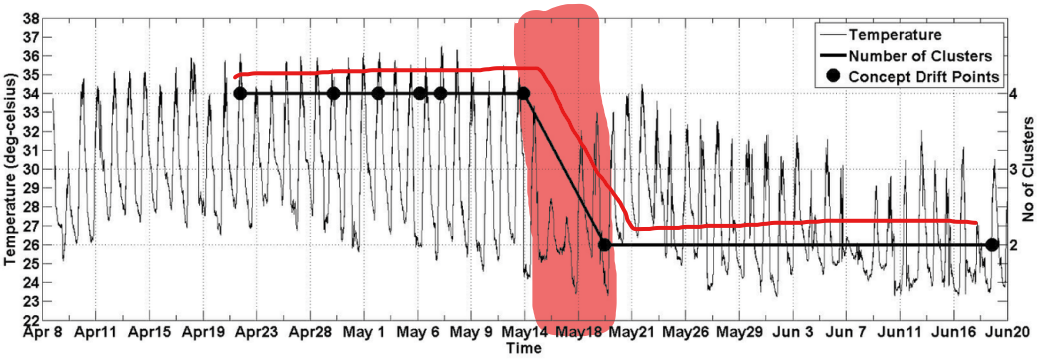
Withdrawal of South-West Monsoon

Figure 4(a) shows the rainfall received during the southwest monsoon withdrawal period. The number of clusters computed at concept drift points are shown in Figure 4(b). Concept changes are found relatively frequent compared to the monsoon onset period. A sudden increase in the number of clusters can be seen in Figure 4(b), corresponding to the two main rainfall events. These results also support the fact that a significant change in clustering is an indication of a significant weather event.

Figure 3. (a) Rainfall during the period 8th April to 20th June. The week of 14th May denotes the start of a pre-monsoon shower. (b) Concept drift points and the corresponding number of clusters estimated for the period 8th April to 20th June. It can be noted that the number of clusters reduced from 4 to 2 from 14th May to 19th May, where the temperature also shows a sudden drop.



(a)



(b)

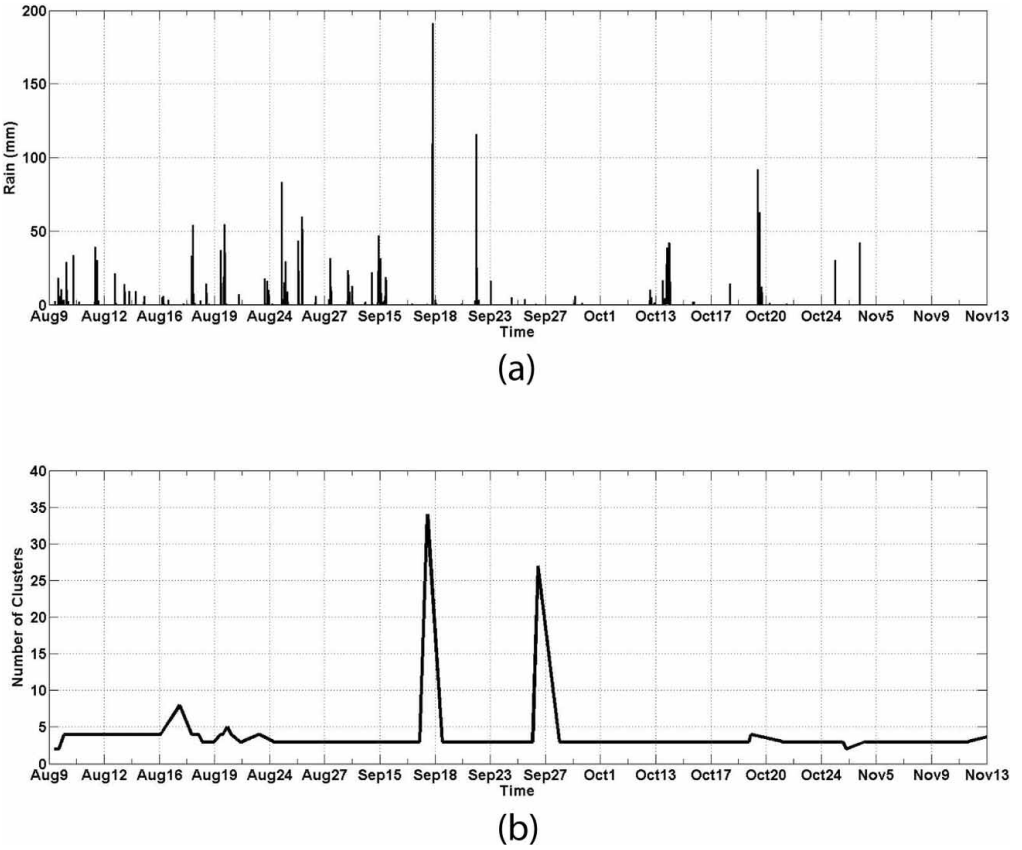
Thunderstorm

In Kerala, during the summer season, it is common to get unpredicted rainfall with thunderstorm and lightning. Weather parameters show considerable changes during thunderstorms as well. Figure 5(a) shows the occurrence of thunderstorms during the summer season. Black dots in the figure denote thunderstorm observations. Temperature, humidity and wind speed are identified to be the most prominent weather parameters that show variations during a thunderstorm. Datastream containing these three parameters is taken to check the clustering structure behaviour and its relation to the thunderstorm occurrence. Change in the clustering structure during thunderstorms is visible from Figure 5(b). No concept changes were identified between the beginning of April and 22nd of April.

Effect of Altering the Page-Hinkley Test Parameters

For each weather phenomenon, the experiments are repeated by changing the Page-Hinkley Test parameters - warning threshold (λ_w), alarm threshold (λ_A) and the tolerance level (δ) – to study their effect on identifying the concept drifts. Some common observations are discussed in this section. As usually done in online algorithms, these parameters are made self-adjusting depending on the average cluster radius. They change value automatically after each concept drift. This is done to ensure that changes are happening to the test parameters according to the recent characteristics of the data. As

Figure 4. (a) Rainfall during the period - August to November; (b) Number of clusters computed for the period - August to November



Silva et al. (2017) and Zhang et al. (2010) suggest, threshold values have to be chosen as a function of the current clustering itself. Fixed threshold values might make it unrealistic to handle an evolving data stream. Value of λ_A is determined by multiplying the current average cluster radius with a fixed real constant. λ_w and δ are made dependent on λ_A. These real constants have to be provided to the algorithm once in the beginning as external parameters.

PHT parameters control the trade-off between early detection of concept drift and false alarms (Gama, 2014). Values of λ and δ can be used to tune the performance of concept drift detector. Lower values of λ cause the concept drift detector to identify even minute changes and often leads to false alarms. Larger values of λ cause less false alarms but some changes might be missed. Similarly, lower the value of δ, faster the change detection will be.

Results of the experiments done on monsoon onset period with different PHT parameters are shown in this section. Figure 3(b) corresponds to the onset of monsoon with $\lambda_A = 1.5$ times the average cluster radius and $\delta = 0.1*\lambda_A$. Effect of increasing the value of λ_A, on the same data is depicted in Figure 6. λ_A is set to 1.75 times the average cluster radius and δ is kept same as $0.1*\lambda_A$. Since the value of λ_A is increased, the smaller number of concept changes are detected shown in Figure 6. Also, the change in the data distribution happened from 15th to 17th May, is detected only after a considerable delay.

Figure 7 shows the effect of changing the value of δ with the same data. When the value of δ is decreased to $0.01*\lambda_A$, change is alarmed in a faster way. Change in distribution that started during 15th May is captured almost on the same day.

Figure 5. (a) Thunderstorm occurrence; (b) Number of clusters computed during the concept drifts

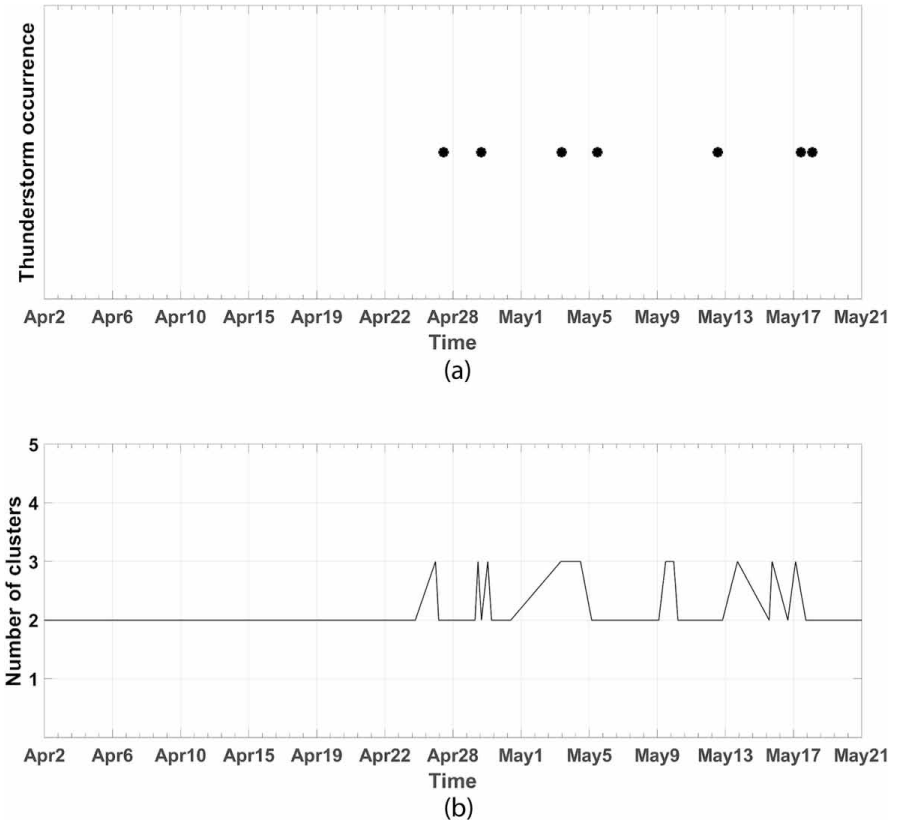
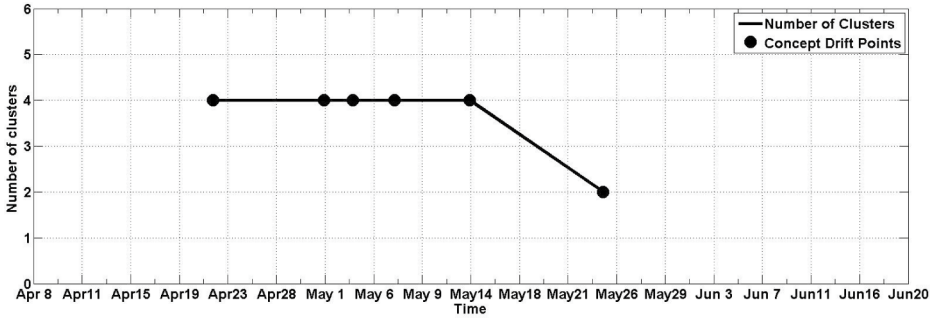


Figure 6. Concept drift points when λ_A = 1.75 times the average cluster radius and $\delta = 0.1*\lambda_A$



Cluster Evolution Analysis

The result of cluster evolution analysis is plotted as a percentage of clusters in the current clustering that have undergone absorption, survival and split. Comparison between the clustering is performed whenever a new clustering is formed after detection of concept drift. From Figure 8, it can be observed that from 29th April to 14th May, the complete set of clusters survived, weather conditions are also stable during this time. There are some notable changes happening during 14th May. Clusters are getting absorbed and split up from 14th May to 25th May. This time period corresponds to the start of

Figure 7. Concept drift points and the corresponding number of clusters estimated when λ_A = 1.5 times the average cluster radius and $\delta = 0.01*\lambda_A$. Since the value of δ is reduced, it reports concept changes impatiently.

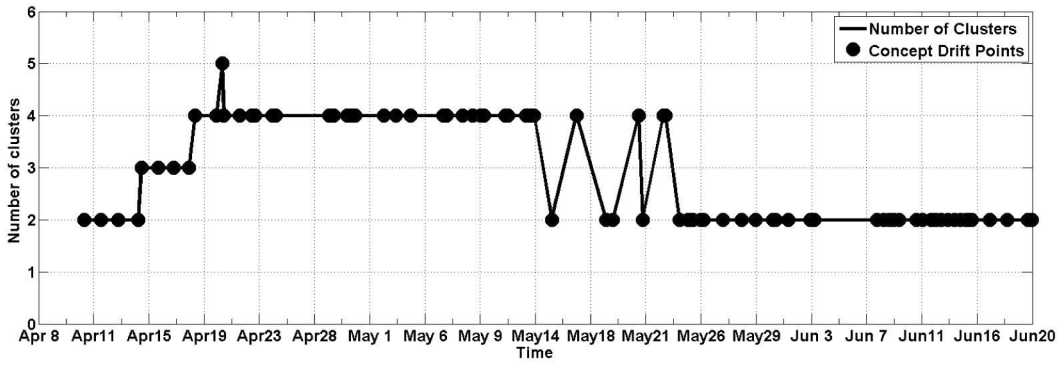
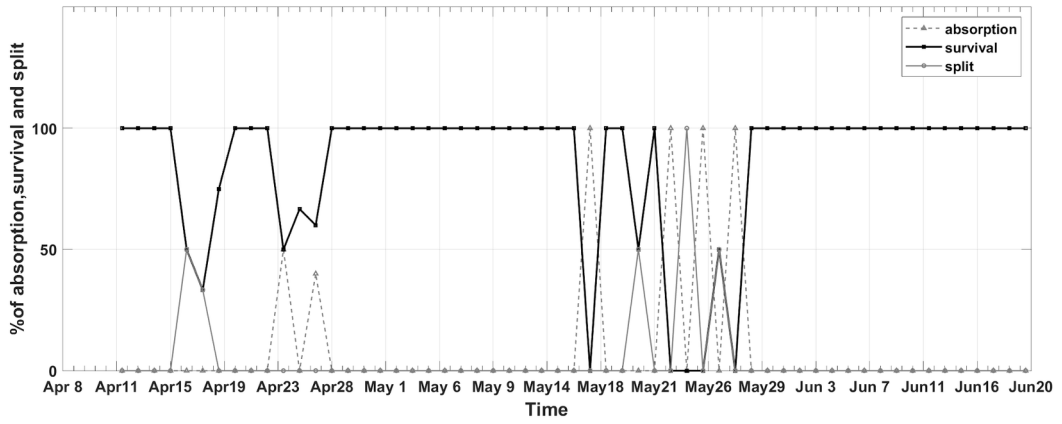


Figure 8. Cluster evolution analysis: Percentage of clusters that have been absorbed, survived and split at consecutive concept change points. This plot corresponds to the concept change points shown in Figure 7, i.e., when λ_A = 1.5 times the average cluster radius and $\delta = 0.01*\lambda_A$.



pre-monsoon shower as evident from Figure 3 (a) and 3(b). Therefore, it can be concluded that the cluster transitions pattern also gives a hint on changing weather conditions.

Comparison With Some Existing Methods

Change point detection is an important topic in time series analysis and prediction. Change point in a time series refers to the time point at which the distribution of the series changes abruptly, necessitating an update to the prediction model (Aminikhanghahi, 2017). Climate data is identified to be one of the important fields where change point detection has good scope (Ducré-Robitaille, 2003; Itoh, 2010; Science, 2006).

The data used in this study can be considered as a time series and hence the change point detection techniques can be applied to it. As far as unsupervised learning is concerned, the change point detection of time series analysis and the concept drift detection of data stream mining, both focuses on identifying changes in the data distribution. But there are some advantages when data is considered as a stream and concept drift detection algorithms are used. These advantages can be listed based on the challenges to be addressed by the change point detection algorithms as discussed in (Aminikhanghahi, S, 2017):

- **Online change detection:** The proposed framework ensures online change detection of the data stream. It can be classified as ε-real-time algorithm as it requires minimum ε number of samples to identify a change. Also, the proposed method supports multiple change point detection;
- **Non-parametric:** The proposed method is non-parametric in nature as it requires no assumptions about the distribution of the data;
- **Support to multi-dimensionality:** The proposed framework supports multidimensional data.

The proposed framework does not require the historical data to be stored in main memory; instead, it just keeps snapshots of data in secondary memory, as it is discussed in CluStream algorithm (Aggarwal et al. 2003). This reduces the memory-related problems considerably and makes the change detection process faster.

The major limitation of the proposed method is that change detection is sensitive to the initial PHT parameters provided to the algorithm. But most of the time-series change point detection algorithms also take external parameters which decide the quality of the change detection – like the maximum number of change points, significance level, minimum window size etc.

To compare the results, the weather data stream is applied to different time series change point detection algorithms. Since the data is multi-dimensional, methods supporting multi-dimensional data are chosen for the comparison. Change point detection algorithms which handle true multi-dimensional data are not plenty in number as most of them split the multivariate data to independent single-dimensional series and detect the changes in marginal distributions. R package named ecp is providing algorithms for multiple change-point detection on multivariate data (James, N.A, 2015). Energy statistic is used as the divergence measure to find the difference between distributions.

The weather data streams used in this work are too long to be applied directly to these change point detection algorithms. As mentioned in (James, N.A, 2015), algorithms in ecp package are not recommended for time series containing more than several thousands of observations. Hence a part of each time series – with 10,000 observations - that corresponds to some important weather events is chosen for comparison. Algorithms e.divisive and e.cp3o_delta of ecp package are applied on the three weather data streams corresponding to monsoon onset, withdrawal and thunderstorm. From the monsoon onset data stream, samples from 50,000 to 60,000 are chosen as this part includes the week of pre-monsoon shower. Similarly, from monsoon withdrawal series, samples from 35000 to 45000 are taken because this part includes a heavy rainfall event. From the thunderstorm series, samples from 70000 to 80000 are chosen as it includes two thunderstorm observations. Results quoted in Table 1. corresponds to the e.divisive algorithm performed with significance level = 0.1, R = 100 and minimum window size = 2000 and e.cp3o_delta algorithm with K=2, alpha=1 and delta=999.

Both e.divisive and e.cp3o_delta algorithms identify distribution changes happening in the time series. Related to the important physical events, the data distributions of the series are changing and these algorithms have captured such changes. For example, in the onset series, both the algorithms have detected change point around 55000 which corresponds to the start of the pre-monsoon shower.

Table 1. Change points identified by e.divisive and e.cp3o_delta algorithms

Data Stream	Change Points	
	e.divisive	e.cp3o_delta
Monsoon onset (Samples from 50000 to 60000)	54927,58344	54935
Monsoon withdrawal (Samples from 35000 to 45000)	37001,39664,41736	41136
Thunderstorm (Samples from 70000 to 80000)	72322,74348,77243	73796

Similarly, in withdrawal series, the change point around 39000 corresponds to the sudden peak in raining. In thunderstorm series, two thunderstorm observations are around 72000 and 78000.

The need for change point detection algorithms in time series analysis is mainly for correcting the prediction models when the distribution changes. Relationship between the identified change points and the physical events is rarely considered as a concern of these algorithms. In addition to concept change detection, the proposed method is interested in exploring these kinds of relationships as well.

DISCUSSION AND FUTURE WORK

Literature includes many examples where weather/climate data is considered as time series and change point detection methods are used to identify the non-homogeneity or steps in the series (Ducré-Robitaille, 2003; Itoh, 2010; Science, 2006). This work discusses an online, unsupervised concept drift detection method and its application on the high-dimensional weather data stream. Page-Hinkley test is the backbone of the concept drift detection proposed in this framework. This section includes some thoughts on improving the change detection process:

- The main limitation associated with Page-Hinkley test is that it requires three external parameters to be provided by the user depending on the domain and the peculiarities of the application. Often this leads to a variation in the quality of the test. Even though the calculation of these parameters is made partially automatic, the other part is still fixed and provided by the user. The change detection improves if this calculation can be fully automatically done from the characteristics of the data. (Sebastião, R, 2017) discusses an approach towards this goal;
- Size of the warning period – the period between the warning signal and the alarm signal - is another important concern. A minimum size criterion helps to ensure that the new model is built from sufficient samples. Similarly, putting a constraint on the maximum size avoids the possibility of warning period growing too long. Deciding optimum limits to the buffer size will improve the change detection and model generation;
- There should be a method to quantify the drifts and categorize them like normal or rare. Currently, all changes above the alarm threshold are treated as same. But quantifying the changes will help to identify abnormal situations and take timely actions;
- Capability to identify the dimension or dimensions which contributed more to the change will necessarily be an improvement to the change detection process. This might give some insights into the process generating the stream or the reason for the change.

All the above-mentioned points lead to the requirement of more research in this field. In addition to this, scalability is an important issue to be considered. An ideal data stream processing algorithm considers the stream to be infinite and hence the length of the stream should not be an issue. But an increase in the dimensionality of the stream might slow down the response.

CONCLUSION

This paper proposes a stream clustering framework which includes explicit concept drift detection and cluster evolution analysis. The number of clusters to be formed is determined dynamically. This framework helps to derive the relationship between the clustering structure changes and the physical events and thus provide information for the prediction of such events. Weather data is chosen to conduct the experiments and check the application of this framework. Even though the stream mining research is advanced to a great extent, its application in weather prediction tasks is not much studied yet. From the experiment results, it could be concluded that the framework is capable of identifying important clustering structure changes and these changes warn significant physical events. Different

parameters of the framework can be tuned to identify changes of different significance. Currently, the framework is limited to just giving a hint on the physical event; accurate prediction is not handled as such. Depending on the application field, the suitable prediction module can be incorporated.

ACKNOWLEDGMENT

The authors thank both the funding agencies; UGC for granting RUSA scheme and DST for awarding PURSE scheme; which were utilised to build the infrastructure needed for this research. The authors also express their sincere thanks to the technical staff and faculty at ACARR (Advanced Centre for Atmospheric Radar Research), Cochin University of Science and Technology, Kerala, India for providing data and supporting this work.

REFERENCES

Aggarwal, C. C., Watson, T. J., Ctr, R., Han, J., Wang, J., & Yu, P. S. (2003). A Framework for Clustering Evolving Data Streams. *Proceedings of the 29th international conference on Very large data bases* (pp. 81–92). Academic Press. doi:10.1016/B978-012722442-8/50016-1

Aminikhanghahi, S., & Cook, D. J. (2017). A survey of methods for time series change point detection. *Knowledge and Information Systems*, *51*(2), 339–367. doi:10.1007/s10115-016-0987-z PMID:28603327

Bhatnagar, V., & Mignet, L. (2009). A Parameterized Framework for Clustering Streams *International Journal of Data Warehousing and Mining*, *5*(1), 36–56. doi:10.4018/jdwm.2009010103

Bifet, A., G. Holmes, B. Pfahringer, P. Kranen, H. Kremer, T. Jansen, & T. Seidl, (2010). MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering. *Proceedings of HaCDAIS 2010*. Academic Press.

Chakraborty, S., N. Nagwani, & L. Dey, (2012). Weather Forecasting using Incremental K-Means Clustering. *International Journal of Biometrics and Bioinformatics*.

Chen, K., & Liu, L. (2006). Detecting the change of clustering structure in categorical data streams. *Proceedings of the Sixth SIAM International Conference on Data Mining* (pp. 504-508). Academic Press. doi:10.1137/1.9781611972764.49

Domingos, P., & Hulten, G. (2000). Mining high-speed data streams. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '00* (pp. 71–80). ACM. doi:10.1145/347090.347107

Ducré-Robitaille, J.-F., Vincent, L. A., & Boulet, G. (2003). Comparison of techniques for detection of discontinuities in temperature series. *International Journal of Climatology*, *23*(9), 1087–1101. doi:10.1002/joc.924

Duncan, A., Keedwell, E., Djordjevic, S., & Savic, D. (2013). Machine learning-based early warning system for urban flood management artificial neural networks for urban flood modelling. *Environmental Modelling & Software*, *22*(September), 3697–3711.

Faria, E. R., Barros, R. C., Gama, J., & Carvalho, A. C. P. L. F. (2012). Improving the offline clustering stage of data stream algorithms in scenarios with variable number of clusters *Proceedings of the ACM Symposium on Applied Computing* (pp. 829-830). ACM. doi:10.1145/2245276.2245437

Fraley, C., & Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, *97*(458), 458. doi:10.1198/016214502760047131

Gama, J. a., P. P. Rodrigues, E. Spinosa, & A. Carvalho, (2010). Knowledge Discovery from Data Streams. Web Intelligence and Security. In *Advances in Data and Text Mining Techniques for Detecting and Preventing Terrorist Activities on the Web* (pp. 125-138). Academic Press.

Gama, J. A. (2014). A Survey on Concept Drift Adaptation. *ACM Comput. Surv.*

Ghesmoune, M., Lebbah, M., & Azzag, H. (2016). State-of-the-art on clustering data streams. *Big Data Analytics*, *1*(1), 13. doi:10.1186/s41044-016-0011-3

Giraud-Carrier, C. (2000). A Note on the Utility of Incremental Learning. *AI Communications*, *13*(4), 1.

Glahn, H. R. (1982). Statistical Weather Forecasting. Academic Press.

Guhathakurta, P. (2006). Long-range monsoon rainfall prediction of 2005 for the districts and subdivision Kerala with artificial neural network. *Current Science*, *90*(6), 773–779.

Hinkley, D. V. (1971). Inference about the change-point from cumulative sum tests. *Biometrika*, *58*(3), 509–523.

Hulten, G., Spencer, L., & Domingos, P. (2001) Mining time-changing data streams. *Proceedings of the seventh ACMSIGKDD international conference on Knowledge discovery and data mining - KDD '01* (pp. 97-106). ACM. doi:10.1145/502512.502529

Hung, N. Q., Babel, M. S., Weesakul, S., & Tripathi, N. K. (2008). An artificial neural network model for rainfall forecasting in Bangkok, Thailand. *Hydrology and Earth System Sciences*, *13*(8), 1413–1425. doi:10.5194/hess-13-1413-2009

Itoh, N., & Kurths, J. (2010). Change-point detection of climate time series by nonparametric method. *Proceedings of the world congress on engineering and computer science* (Vol. 1, pp. 20-23). Academic Press.

James, N. A., & Matteson, D. S. (2015). ecp: An R package for nonparametric multiple change point analysis of multivariate data *Journal of Statistical Software, 62*(7), 1-25.

Krasnopolsky, V. M., & Fox-Rabinovitz, M. S. (2005). Complex hybrid models combining deterministic and machine learning components as a new synergetic paradigm in numerical climate modeling and weather prediction. *Proceedings of International Joint Conference on Neural Networks* (Vol. *3*, pp. 1615–1620). Academic Press.

Kuligowski, R. J., & Barros, A. P. (1998). Localized Precipitation Forecasts from a Numerical Weather Prediction Model Using Artificial Neural Networks. *Weather and Forecasting*, *13*(4), 1194–1204. doi:10.1175/1520-0434(1998)013<1194:LPFFAN>2.0.CO;2

Leskovec, J., Rajaraman, A., & Ullman, J. (2014). Clustering. In *Mining of Massive Datasets* (pp. 240–280). Cambridge University Press.

Little, M., McSharry, P., & Taylor, J. (2009). Generalized linear models for site-specific density forecasting of UK daily rainfall. *Monthly Weather Review*, *137*(September), 1031–1047.

Liu, Y., Shi, J., Yang, Y., & Lee, W.-J. (2012). Short-Term Wind-Power Prediction Based on Wavelet Transform-Support Vector Machine and Statistic-Characteristics Analysis. *IEEE Transactions on Industry Applications*, *48*(4), 1136–1141. doi:10.1109/TIA.2012.2199449

Meza, F. J., Hansen, J. W., & Osgood, D. (2008). Economic value of seasonal climate forecasts for agriculture: Review of ex-ante assessments and recommendations for future research. *Journal of Applied Meteorology and Climatology*, *47*(5), 1269–1286. doi:10.1175/2007JAMC1540.1

Mouss, H., Mouss, D., Mouss, N., & Sefouhi, L. (2004). Test of page-hinckley, an approach for fault detection in an agro-alimentary production system. *Proceedings of the 5th Asian Control Conference* (pp. 815–818). Academic Press.

Naldi, M. C., Fontana, A., & Campello, R. J. G. B. (2009). Comparison among Methods for k Estimation in k-means. *Proceedings of ISDA 2009 - 9th International Conference on Intelligent Systems Design and Applications* (pp. 1006–1013). Academic Press.

Nayak, M.A., & Ghosh, S. (2013). Prediction of extreme rainfall event using weather pattern recognition and support vector machine classifier. *Theoretical and Applied Climatology, 114*(3-4), 583–603.

Ntoutsi, I., Spiliopoulou, M., & Theodoridis, Y. (2009). Tracing cluster transitions for different cluster types. *Control and Cybernetics*, *38*(1), 239–259.

Oliveira, M., & Gama, J. (2010). MEC–Monitoring Clusters' Transitions. *Proceedings of the 2010 conference on STAIRS*. Academic Press.

Page, E.S. (1954) Continuous Inspection Schemes. Biometrika, 41, 100-115.

Radhika, Y., & Shashi, M. (2009). Atmospheric Temperature Prediction using Support Vector Machines. *International Journal of Computer Theory and Engineering*, *1*(1), 55–58. doi:10.7763/IJCTE.2009.V1.9

Raghavendra, S., & Deka, P. C. (2014). Support vector machine applications in the field of hydrology: A review. *Applied Soft Computing*, *19*, 372–386. doi:10.1016/j.asoc.2014.02.002

Rajeevan, M., Pai, D. S., Dikshit, S. K., & Kelkar, R. R. (2004). IMD's new operational models for long-range forecast of southwest monsoon rainfall over India and their verification for 2003. *Current Science*, *86*(3), 422–431.

Reeves, J., Chen, J., Wang, X. L., Lund, R., & Lu, Q. Q. (2007). A review and comparison of changepoint detection techniques for climate data. *Journal of Applied Meteorology and Climatology*, *46*(6), 900–915. doi:10.1175/JAM2493.1

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

Sakamoto, Y., Fukui, K. I., Gama, J. a., Nicklas, D., Moriyama, K., & Numao, M. (2016). Concept Drift Detection with Clustering via Statistical Change Detection Methods. *Proceedings - 2015 IEEE International Conference on Knowledge and Systems Engineering, KSE 2015* (pp. 37–42). IEEE Press.

Sebastião, R., & Fernandes, J. M. (2017, June). Supporting the Page-Hinkley test with empirical mode decomposition for change detection. *Proceedings of the International Symposium on Methodologies for Intelligent Systems* (pp. 492-498). Cham: Springer.

Serakiotou, N. (1987). Change detection.

Silva, J., Faria, E. R., Barros, R. C., Hruschka, E. R., Carvalho, A. C. P. L. F. D., & Gama, J. (2013). Data stream clustering. *ACM Computing Surveys*, *46*(1), 1–31. doi:10.1145/2522968.2522981

Silva, J., & Hruschka, E. R. (2011). Extending k-means-based algorithms for evolving data streams with variable number of clusters. *Proceedings - 10th International Conference on Machine Learning and Applications, ICMLA 2011* (Vol. 2, pp. 14–19). Academic Press.

Silva, J. D., Hruschka, E. R., & Gama, J. (2017). An evolutionary algorithm for clustering data streams with a variable number of clusters. *Expert Systems with Applications*, *67*, 228–238. doi:10.1016/j.eswa.2016.09.020

Sivakumar, M. V. K. (2006). Climate prediction and agriculture: Current status and future challenges. *Climate Research*, *33*, 3–17. doi:10.3354/cr033003

Smith, B., Hoogenboom, G., & McClendon, R. W. (2009). Artificial neural networks for automated year-round temperature prediction. *Computers and Electronics in Agriculture*, *68*(1), 52–61. doi:10.1016/j.compag.2009.04.003

Spiliopoulou, M., Ntoutsi, I., Theodoridis, Y., & Schult, R. (2006, August). Monic: modeling and monitoring cluster transitions. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 706-711). ACM.

Vendramin, L., Jaskowiak, P. a., & Campello, R. J. G. B. (2013). On the combination of relative clustering validity criteria. *Proceedings of the 25th International Conference on Scientific and Statistical Database Management - SSDBM*. Academic Press. doi:10.1145/2484838.2484844

Wang, D., & Cai, X. (2009). Irrigation Scheduling Role of Weather Forecasting and Farmers Behavior. *Journal of Water Resources Planning and Management*, *135*(5), 364–372. doi:10.1061/(ASCE)0733-9496(2009)135:5(364)

Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., & Petitjean, F. (2016). Characterizing concept drift. *Data Mining and Knowledge Discovery*, *30*(4), 964–994. doi:10.1007/s10618-015-0448-4

Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning*, *23*(1), 69–101. doi:10.1007/BF00116900

Wilks, D. S. (2006). Statistical methods in the atmospheric sciences (Vol. 59). Academic Press.

Zhang, X., Germain, C., & Sebag, M. (2010). Adaptively detecting changes in Autonomic Grid Computing. *Proceedings - IEEE/ACM International Workshop on Grid Computing* (pp. 387–392). Academic Press. doi:10.1109/GRID.2010.5698017

Namitha K. has received the B.Tech Degree in Computer Science and Engineering from Calicut University, Kerala, India in 2006 and M.Tech Degree in Software Engineering from Cochin University of Science and Technology, Kerala, India in 2008. She has worked as a senior software engineer in Honeywell Technology Solutions Lab, Bangalore from 2008 to 2013. She is currently pursuing her PhD Degree with Cochin University of Science and Technology, Kerala, India. Her research interests include machine learning, data stream mining and big data.

Santhosh Kumar G. is a professor at Cochin University of Science and Technology, India. He is with AI & Computer Vision lab at the Department of Computer Science. His area of interests includes computer vision, cyber-physical systems, and natural language processing.