Concept Drift Detection Technique using Supervised and Unsupervised Learning for Big Data Streams

Manzoor Ahmed Hashmani1[1,2,3]
Syed Muslim Jameel[3†], Vali Uddin4 and Syed Sajjad Hussain Rizvi[4]

[1]High Performance Cloud Computing Center (HPC3), Universiti Technologi PETRONAS, Malaysia
[2]Center for Research in Data Science (CERDAS), Universiti Technologi PETRONAS, Malaysia
[3]Department of Computer and Information Sciences, Universiti Technologi PETRONAS, Malaysia
[4]Faculty of Engineering Sciences and Technology, Hamdard University, Pakistan

Summary
Big Data Stream Analysis (BDS) has a pivotal role in the current computing revolution. The BDS possesses dynamic and continuously evolving behavior and may cause a change in data distribution arbitrarily over time. The phenomenon of change in data distribution over time is known as Concept Drift (CD). CD issue makes classical Machine Learning (ML) approaches in-effective, and ML approaches to need to be adapted to such change to maintain their performance accuracy. Also, CD detection and mitigation are two critical issues. Whereas, CD detection is a crucial pre-requisite of its mitigation, which aims to characterize and quantify CD by identifying change points from the Big Data input stream. Cur-rent CD detection techniques are based on Statistical Analysis and Data Distribution Analysis. However, these approaches do not provide a satisfactory way to differentiate between the concept of drift and noise. Furthermore, in the existing CD detection techniques, the optimize detection time and minimize the error rate is essential. Therefore, this research aims to propose a computational and performance effective concept drift approach. The proposed approach is divided into two modules Unsupervised and Supervised. In the Unsupervised module, the training data is clustered using K-Mean clustering, and the distance between their Centroids are compared with input data using the Cosine Distance. Whereas, in the Supervised module, the classification is performed using the ANN model. Later, the output observed from the Unsupervised and Supervised approaches makes the proposed model very advantageous (accurate). In this paper, we presented some initial experiments to determine Clusters and Centroids points, here we also find out the similarity between the Centroids and input data sample using the Cosine Distance formula. Finally, we did some experiments for the classification module to figure out the optimized classifier for the classification module. In future work, we will validate our proposed solution using the Synthetic and Real Concept Drifted Big Data Streams.

Key words:
Big Data Analysis, Non-Stationary Environment, Drift Detection.

1. Introduction

The Big Data in nature is very sophisticated and versatile. Whereas, the Variability and Veracity characteristics of Big Data are unpredicted. Due to which the Intelligent Systems (use Machine Learning at care, such as prediction, clustering or classification) unable to adjust these dynamic behaviors of Big Data, such as Concept Drift issue [1]. The adaptability in Intelligent Systems is essential to mitigate the Concept Drift issue. These dynamic approaches can be categories into partial and fully dynamic [2]. The concept adaptability in Intelligent Systems is to make classification or prediction models self-regulate, which will adjust the Intelligent Systems dynamically if the data trends change (data trends change due to Variability and Veracity) from the input data streams. Concept drift detection is a pre-requisite step of its handling. Primarily, the Concept drift detection process involves the characterization and quantification of possible changes in the input stream. Jie Lu et al. [3] discuss the concept drift detection approach and established a four-step framework, which includes 1) Data Retrieval 2) Data Modeling 3) Test Statistics Calculation and 4) Hypothesis Testing.

The data retrieval process divides the input streams into different chunks and compares the various fragments to identify their patterns. Whereas the data moving process reduces the data dimensionality and selects the useful features, this step is more useful in high dimensional data-streams, and can significantly minimize the computational cost. Test statistics calculation measure the dissimilarity between the obtained pattern; fundamentally, this step identifies the potential concept drift by using different distance metrics such as Manhattan distance, Averagedistance, Chorddistance, and Eudiclean distance [4]. However, the hypothesis testing is performed to validate the obtained results using test statistics, bootstrapping, the permutation test, and Hoeffding’s inequality-based bound identification [5]. The type of CD (virtual, real, hybrid) and nature of input stream (numeric, text, or imagery) are two principal factors to be noticed for CD detection.

Furthermore, the problem of how to define an accurate and robust dissimilarity measurement for Concept Drift detection is still an open research question. It is challenging because in steaming data analysis,

Manuscript received January 5, 2020
Manuscript revised January 20, 2020

identification of correct input labels is critical and supervised learning is not feasible. Under considering this statement as a hypothesis, this study aims to investigate the data distribution-based Concept Drift detection using Unsupervised Learning. Initially, several datasets (IRIS, Employee, Customer) dataset are investigated using the different configurations of K-Mean clustering. The propose of these experiments is to measure potential class boundaries without the proposal of Concept Drift detection.

2. Related Work

Generally, the classification models are divided into two significant types, such as Offline or Batch Learning and Online or Real-Time Learning. In offline mode, the behavior input data for classification is deterministic, and the feature of class distribution of input data will always be the subset of the training dataset. However, in online learning, particularly real-time stream classification using data streams are not deterministic. In big data streams classification, the uncertainty in input streams changes the input feature behavior and class distribution, which in return causes a massive deterioration in the classification algorithm, which significantly affects the performance. To detect these feature changes from the input streams during online learning is one of the potential research questions from last decay. These proposed techniques are sequential based on Statistical Analysis, such as Sequential Analysis [6, 7], Control Charts [8, 9], and Data Distribution Analysis, such as Monitoring two distributions [10, 11] or Contextual approaches [12, 13].

The Statistical Analysis (SA) observes the probability of or error during online streaming observations, such as classification [14]. Statistical Process Control and Control charts are two potential examples of SA. Drift Detection Method [15], is one of the initially proposed techniques for CD detection. DDM technique is based on Statistical Process Control, which determines the potential CD by observing the classifier performance. This approach fixed a threshold level (warning level) of the allowable performance of the classifier; if the classifier exceeds the warning level, then that condition is declared as potential Concept Drift. This approach has a good behavior detect-ing abrupt changes and gradual changes when the progressive change is not very slow, but it has difficulties when the switch is slowly step-by-step [14]. Later, based on the DDM techniques various other CD detection techniques introduced, such as the Early Drift Detection Method (EDDM) [16] in order to solve the problem associated with DDM. In the SA, the primary concern is to differentiate the issue of misclassification. For example, if the model exceeds the warning threshold level due to low performance. To cope with these issues, researchers investigate data distribution de-spite the classifier performance itself. For example, They are monitoring Data Distribution. This technique typically uses a fixed reference window that summarizes the past information, and a sliding detection window over the most recent examples [17]. The core idea is comparing two distributions over two windows using statistical tests with the null hypothesis stating that the distributions are equal. If the null hypothesis is rejected, a concept drift is declared — adaptive windowing algorithm (ADWIN) [18]. The computational cost and memory cost in the Data Distribution based CD techniques are two primary concerns. Through the comprehensive literature analysis on the CD detection techniques, we can safely state that still the concept drift detection is not deterministic, and several limitations can be highlighted from the proposed meth-ods. For example, the difference between the actual CD and noise, the existing solutions cannot learn from the multiple concepts, the need a massive amount of data to be analyzed the drift pattern. Therefore, this study is a step towards the proposal of Concept Drift detection techniques to somehow minimize these limitations.

3. Methodology

In the proposed approach, the two different modules (Unsupervised and Supervised) provide their results individually; later, the submitted score of those modules calculate the confidence to predict the Concept Drift. In the Unsupervised module, we have employed the clustering of training data by using k-means clustering. Later, K-means clustering also calculates the centroids of the data. Afterward, cluster-based data is mapped with the input data sample using Cosine Distance. Here we define a distance threshold value T (50%). If the value exceeds the T, then it will the given a vote for potential Concept Drift. Moreover, in the Supervised module, the same input sample is also classified using the ANN model, if the classification accuracy obtained from the ANN model (trained using training dataset) is less than 50% for each class (in multiclass classification), then it will the given a vote for potential Concept Drift. Finally, If both modules categorize input data sample as likely Concept Drift, then that input will be detected as Concept Drift, as shown in Fig. 3.

3.1 Dataset

SEA dataset: SEA dataset is used for annotation of concept drift. Firstly, both datasets are distributed in training and testing datasets. The SEA dataset contains 60000 examples, while Stagger has 100000 instances. SEA datasets have three (03) features. The dataset is separated into two cases are training, and testing using k means clustering (as shown in Fig. 17,18). SEA dataset contains 50000 for training and 10000 for testing. While stagger dataset contains 80000 for training and 20000 for testing.

IRIS dataset: The IRIS data set contains three (03) classes of 50 instances each, where each class refers to a type of iris plant. One lesson is linearly separable from the other, the latter are NOT linearly separable from each other, as shown in Fig. 1. Therefore, in our experiments, we have used only the two attributes, such as petal length and petal width.

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

Fig. 1 IRIS dataset and the values of its features

Employee Dataset: In order to visualize the more complex clustering scenario, we have created our own datasets, containing the 500 different records. This dataset include the two main features (Salary and Age) to be analyzed, as shown in Fig. 2. The attribute/ feature values mentioned in the provided dataset are so tightly coupled, which makes the clustering more challenging.

	name	age	salary
0	manzoor	26	20644
1	ahmed	21	186846
2	hashmani	45	39734
3	syed	42	134124
4	muslim	22	34155

Fig. 2 Employee dataset and its contributor features (age and salary) for clustering.

3.2 Tools and Techniques

The provided results in this study are performed using Python 3 and its API (Tensorflow 1.13, Keras 2.02. The training for clustering and classification took place in Google Colaboratory GPU in the Colab Jupyter Notebook. The various configurations of K-Mean clustering algorithm with the ELBOW technique are used for clustering, whereas for classification Tree, Discriminant Analysis, Support Vector Machine, KNN, and Rusboosted classifiers are used, as shown in Appendix (Table 1).

3.3 Proposed Algorithm

Input: Multiple data-sources DS: (ds1, ds2, …., dsn), Continuous data inputs D: (DS1, DS2,…….DSn) at time =t and D: (DS1, DS2,…….DSn+1) at time =t+1 (Concept Drift). T is the training space (T1, T2,……Tn), Tn is the number of training samples. C is the centroid of the clusters obtain from training samples (C1, C2………, Ci), CB is the Cluster boundary (CB1, CB2,………CBj), and T is threshold value of the classification performance (T =0-.5) Concept Drift using Traning dataset (CD1) and Concept Drift using classifier (CD2).

Output: Detected Concept Drift (CD) time (identify the new spectral band at time: t+1).

1: Take the input data sample from the input stream (DS)

2: Determine the similarity index of the input sample

3: Compute the clusters
  3.1 Place the random centroid point (Ci) (K=3)
  3.2 Compare the training sample points (Tn)
//To determine the nearest neighbor with the centroids
//using distance function (Eudiclean Distance).
  3.3 Update the Centroid (Ci)
// By taking average (T1+T2+…..+Tn)/n.
4: Compute the input data (DS) similarity index (Si)

5: Compare the Si with the Ci
  5.1: if (Si ==Ci with range Tn)
  5.2: Set Concept Drift (CD1)=0
  5.3: Else CD1=1

6: Compute the accuracy of the classifier (F(X)=f(Ewx+b)<=T)
  6.1: if F(X)> T then set Concept Drift (CD2) =0
  6.2: Else CD=1

7: If CD1=CD2 then raise alarm for Concept Drift detection.

4. Result and Discussion

In this research paper, we followed two initial experiments towards the implementation of our proposed algorithm. Firstly, we performed simulations for the Unsupervised module, and here we figured out the clusters by using the K Mean algorithm and monitored the changes of centroid points after each cluster (for example, initially, we observed the centroid values of 2 clusters, then 3 clusters till 12 clusters). Also, we highlighted the issue of dynamical selection of the number of clusters. In these experiments, we used the ELBOW method to see the appropriate number of clusters. Later in the first experiment, we identify the outlier anomalies (the point which is not relevant to the particular classes), in future we intend to compare the detected real-time outlier anomalies by applying the cosine distance between the input sample data point (input data from input stream) and centroids of the acquired clusters values. Secondly, we did some initial experiments for the Supervised module, here we tested 22 state-of-art classifiers and checked the most appropriate and suitable classifier for Concept Drift detection.

4.1 Experiment 1: Investigation to identify the optimal number of clusters and determine clusters centroid values for Concept Drift Detection in Unsupervised Learning Module

Initially, we visualized and clustered the SEA dataset (Concept Drifted dataset) for the two relevant attributes at1, at2, and at3, as shown in Fig 4. Here the visualization of the SEA data represents dense due to its three attributes and is not normalized.

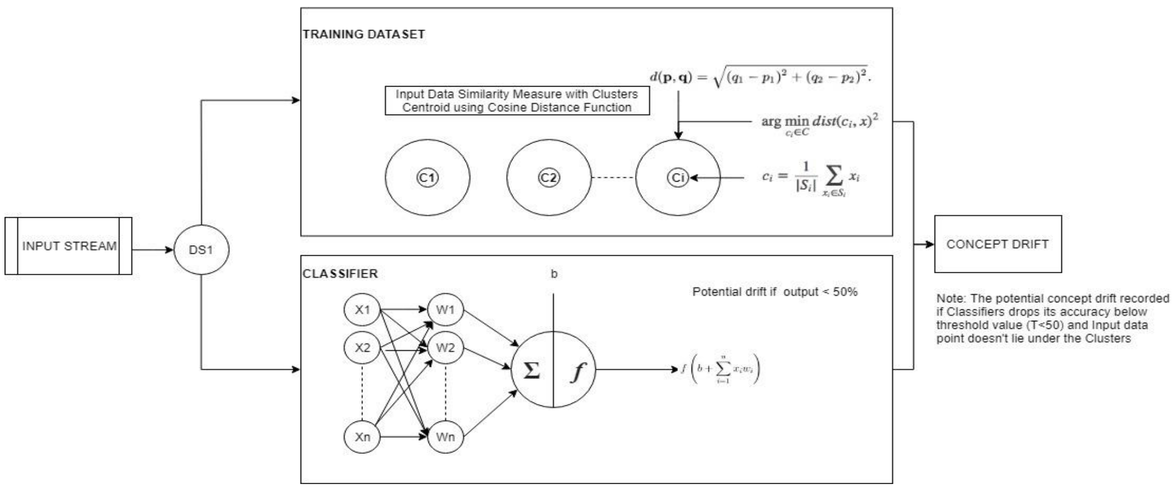

Fig. 3 Concept Drift Detection using the Supervised and Unsupervised Learning (DS1 represents the input data sample, Ci centroid of ith clusters, Xn inputs to the classifier, Wn weights given to each X, b is BIAS value.

Later, we applied the K-Mean algorithm to determine the centroid positions of at1 and at2 attributes, as shown in Fig. 5. The centroids points for the at1 and at2 are far from the actual data clusters points. Therefore we can conclude that realizing the values of the cluster from the drifted dataset is not appropriate, and the better solution is to verify the actual labels using the stable datasets.

In later experiments, we took the example of the IRIS dataset (not drifted), which possesses the high cohesion (among the similar data points) and less coupling (among the three classes). By applying the K-Mean clustering algorithm we figured out the potential number of clusters, where we kept the value of K=2, the results demonstrate the two possible clusters but one missing data sample (a red data sample in the green data sample class) as shown in Fig.5 and Fig. 6., this is due to the data is not normalized, later we applied 0-1 normalization and then verified the clusters with the same value of K=2 and we observed a significant betterment in the clustering performance (both clusters are more appropriate and do not perform misclustering), as shown in Fig. 8 and Fig. 9. Additionally, we also tested normalized IRIS datasets with the value of K=3 (because the IRIS dataset contains three classes) and visualized their results, as shown in Fig.10 and Fig. 11. However, to determine the optimal number of clusters we used the ELBOW method and kept the Range=1 to 10. Interestingly, the most appropriate number of clusters is observed when we selected the value of K=2. However, K=3 (which are the actual number of classes) were not optimal selection. Through this experiment, we can conclude that the number of clusters is difficult to be correctly identified if the two types have similar features, and ELBOW methods do not perform well to predict it, as shown in Fig. 11.

Finally, we performed some experiments on the more challenging dataset, and we intended to monitor the behavior of K-Mean Cluster and ELBOW methods under a not feasible environment (challenging for Elbow methods). Therefore, we tested these algorithms in our own created dataset (EMPLOYEE), Employee dataset is challenging enough to be clustered because the data points are much relevant to each other, this scenario is very crucial for ELBOW methods to be analyzed. In the employee dataset, the two most appropriate features were selected (age and salary) for analysis purposes. Initially, the employee dataset plotted, visualized, and clustered (with K=3) without normalization. The clustered results were not satisfactory. Furthermore, the locations of centroids were also not correct position, as shown in Fig 12. and Fig. 13. After the normalization process, the clustering results showed better performance with more centered centroids as shown in Fig. 14 and Fig.15. And Table. 3. In order to verify the ELBOW methods, we check the various configurations of clusters using K-means algorithms (K=1 to 7) and visualize the best pattern. Here we figured out (by our observation Fig 15 to Fig.20.). The clusters seem more suitable when the number of clusters is four (K=4). Later we validated our observation by applying the ELBOW method (K range from 1-12), and the Elbow predicted the best cluster when the number of clusters is four (4), K=4, as shown in Fig. 20. Also, we monitored all the changes in centroid after each clustering configuration (mentioned in Table. 3). The obtained centroids values will be used to compute the distance between the similarity of input data (using cosine distance formula) to detect the potential Concept Drift.

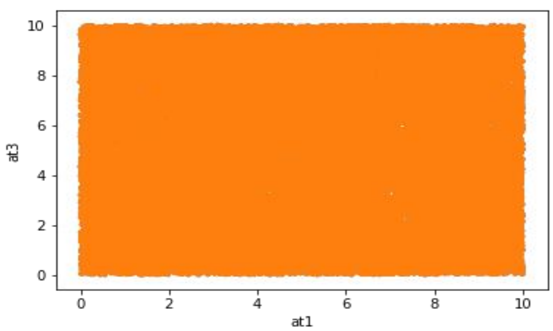

Fig. 4 Visualization of SEA dataset (at1 and at3 attributes)

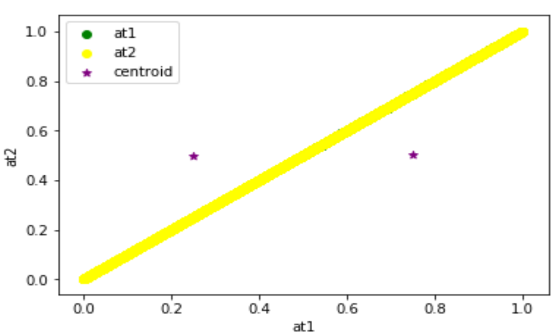

Fig. 5 Clustering of SEA dataset using K-mean technique (K=2)

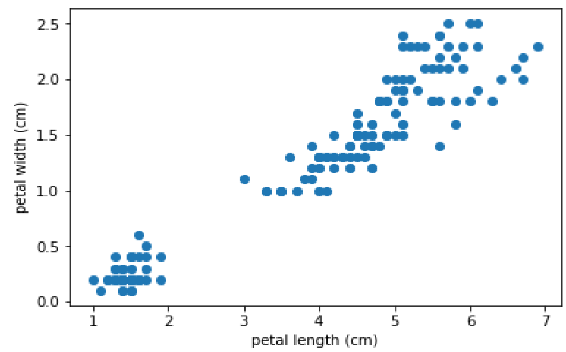

Fig. 6 Visualization of the IRIS datasets with petal attributes

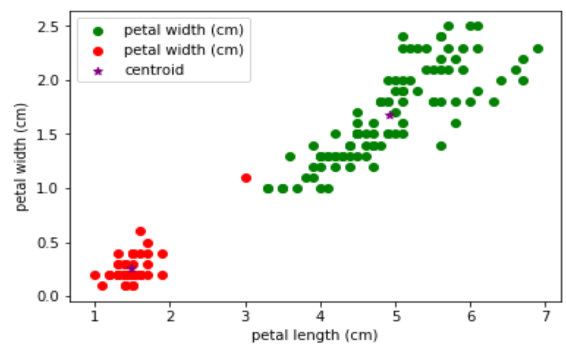

Fig. 7 Clustering of IRIS dataset using K-means technique (K=2)

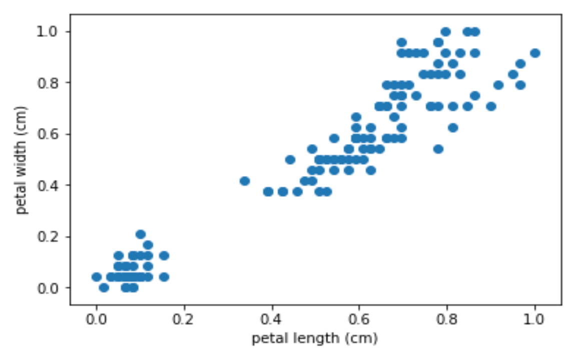

Fig. 8 Visualization of normalized IRIS datasets with petal attributes

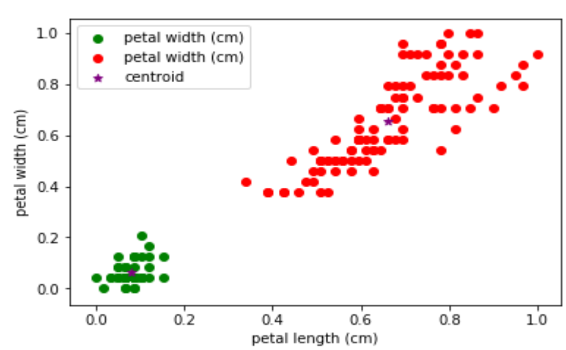

Fig. 9 Clustering of normalized IRIS dataset using K-means technique (K=2)

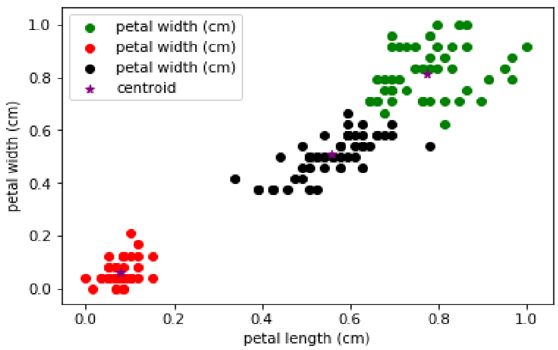

Fig. 10 Clustering of IRIS(norm) dataset using K-means (K=3)

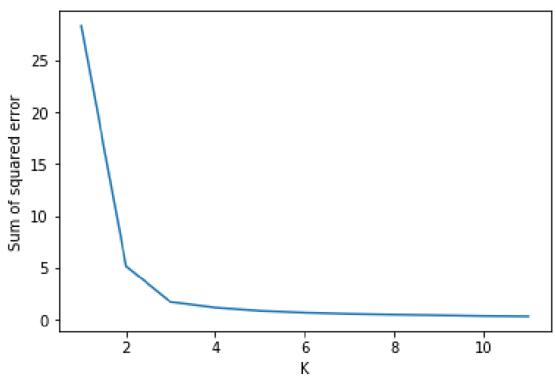

Fig. 11 Elbow representation of K-mean (K range 1-12)

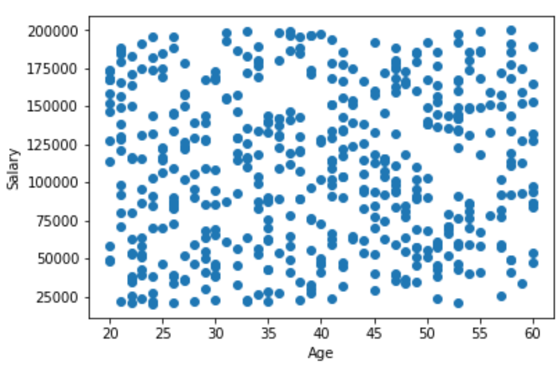

Fig. 12 Visualization of employee dataset

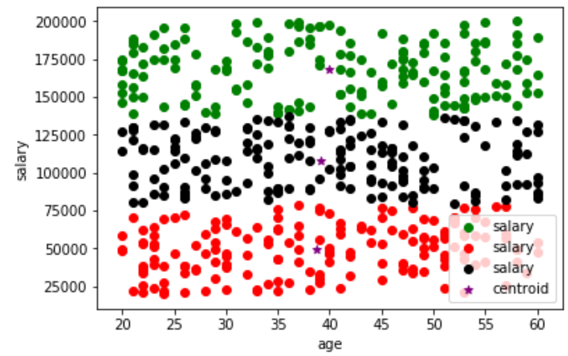

Fig. 13 Clustering of employee dataset using K-mean technique (K=3)

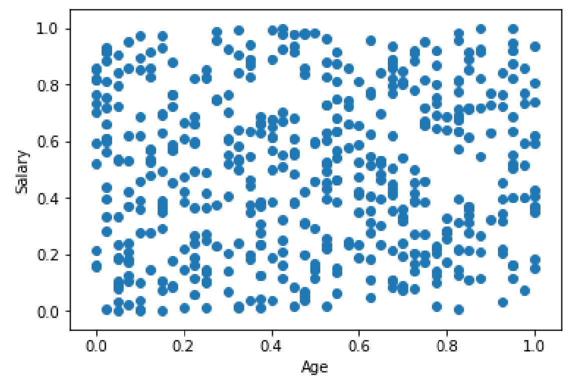

Fig. 14 Visualization of normalized employee dataset

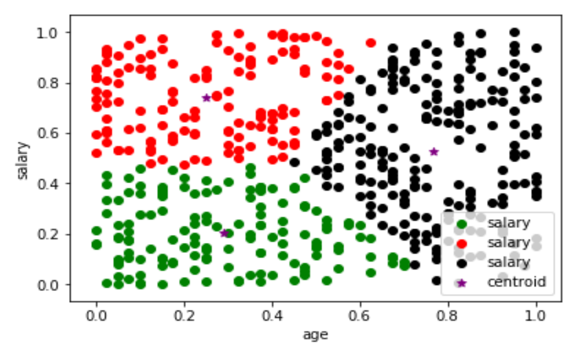

Fig. 15 Clustering of normalized employee dataset using K-mean technique (K=3)

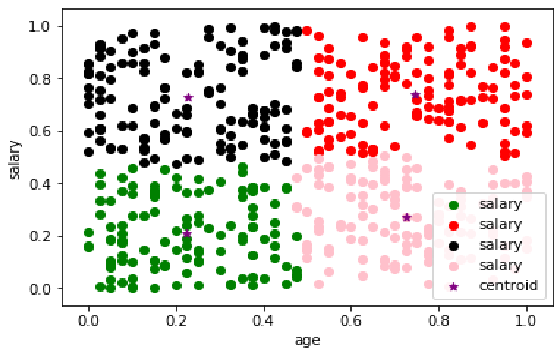

Fig. 16 Clustering of normalized employee dataset using K-mean (K=4)

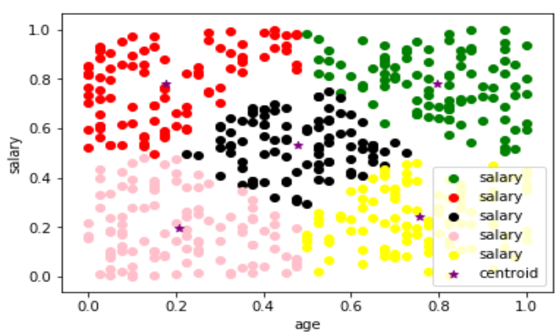

Fig. 17 Clustering of normalized employee dataset using K-mean (K=5)

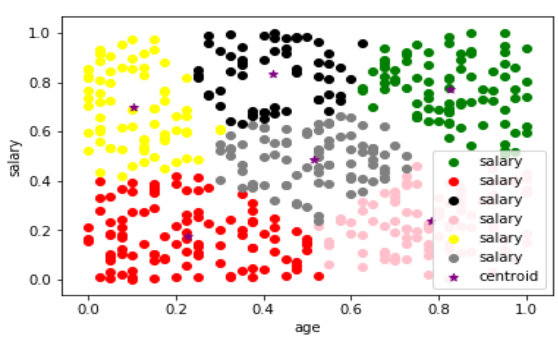

Fig. 18 Clustering of normalized employee dataset using K-mean (K=6)

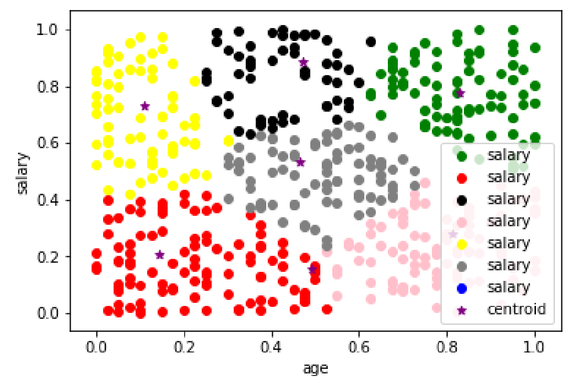

Fig. 19 Clustering of normalized employee dataset using K-mean (K=7)

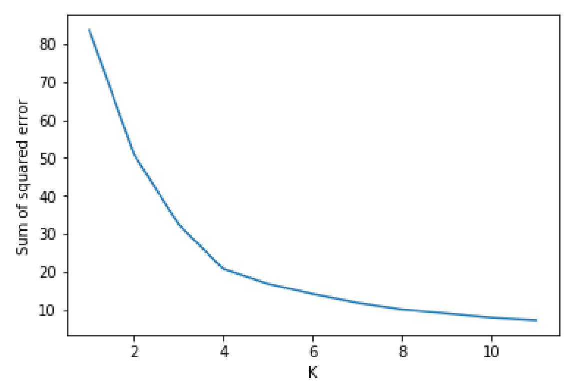

Fig. 20 Elbow representation of K-mean (K range 1-12) clustering using normalized employee dataset

Investigation to detect the Outliers through the anomaly detection for Concept Drift

Anomaly Outlier anomaly exposure is one way to discover the data points out of the boundary of the cluster. This experiment is essential to diagnose the potential Concept Drift. Here the input data sample will be taken to predict the outlier anomaly detection. We aim to analyze the given dataset so we can detect abnormal data. The Iris-Species data is perfect for anomaly detection because it is a clear and complete structure, but also because every species has the same amount of given data. For our analysis, we want to use the Gaussian Mixture Model. This model is convenient for our aim to detect abnormal data and to make predictions of the species per plant. It combines several multivariate normal distributions. However, if the data is hugely unclean, for example, half of the information is an ‘anomaly,’ then it is difficult to identify the anomalies. For example, if we have a dataset which forms two clusters and the data point away from these two clusters can be classified as anomalies. However, if we have many defects that they end up making their cluster, then it will become tough to detect them as outliers. There are various kinds of Unsupervised Anomaly Detection methods such as Kernel Density Estimation, One-Class Support Vector Machines, Isolation Forests, Self-Organizing Maps, C Means (Fuzzy C Means), Local Outlier Factor, K Means, Unsupervised Niche Clustering (UNC) and others.

Table 3: The obtained Centroids coordinates from the employee dataset observed Clusters

Data set	Number of Clusters	Centroids Coordinates Values
Employee Dataset	Three clusters (K=3)	c1(x,y)=[3.99277108e+01,1.67873127e+05] c2(x,y)=[3.91597633e+01, 1.08166089e+05] c3(x,y)=[3.87865854e+01, 4.93527683e+04]
	Three clusters (K=3) (With Normalization)	c1(x,y)=[0.25106383, 0.74119817] c2(x,y)=[0.76690141, 0.52583157] c3(x,y)=[0.28913793, 0.20219151]
	Four clusters (K=4)	c1(x,y)=[0.21688596, 0.2059932] c2(x,y)=[0.75360169, 0.75219524] c3(x,y)=[0.71123188, 0.28522587] c4(x,y)=[0.22383721, 0.73042947]
	Five clusters (K=5)	c1(x,y)=[0.22254464, 0.18651372] c2(x,y)=[0.77272727, 0.79449025] c3(x,y)=[0.775 , 0.25958089] c4(x,y)=[0.46805556, 0.53016052] c5(x,y)=[0.16277174, 0.77233416]
	Six clusters (K=6)	c1(x,y)=[0.20970874, 0.18247757] c2(x,y)=[0.84605263, 0.76308394] c3(x,y)=[0.11153846, 0.71436326] c4(x,y)=[0.75607477, 0.24079795] c5(x,y)=[0.48005618, 0.52650112] c6(x,y)=[0.48804348, 0.88467284]
	Seven clusters (K=7)	c1(x,y)=[0.45141509, 0.16114082] c2(x,y)=[0.48607955, 0.5318344] c3(x,y)=[0.112 , 0.72497357] c4(x,y)=[0.84605263, 0.76308394] c5(x,y)=[0.48804348, 0.88467284] c6(x,y)=[0.80588235, 0.25320773] c7(x,y)=[0.13585526, 0.20923605]

4.2 Experiment 2: Investigation to find the optimal classifier for Concept Drift detection in Supervised Learning module

This study is in the continuation of our previous research paper published [19]. In that paper, among the 22 classifiers, we figure out the performance in the concept drift scenario. In continuation of that research, this study aims to investigate these models further and check their feasibility work as the classifier for Concept Drift detection. Here we want to investigate the performance to negate the possible change of overfitting and underfitting issue, which could cause the potential error during Concept Drift detection.

The support vector machine has minimum training accuracy 75.7, and maximum training accuracy has a complex tree, linear, quadratic, median and coarse Gaussian support vector machine was 87.4 %. Also, RUS Boosted maximum testing accuracy was 84.9433, while minimum testing accuracy was 37.3067 of the elaborate tree, linear SVM and median SVM. Furthermore, the peak prediction speed was detected in quadratic ratio discriminant while lowest prediction speed in cosine KNN. To sum up, the RUS Boosted tree model was found in the best model, as shown in Appendix (Table 2).

Through the analysis of the obtained results, we can suggest using the RUS-Boosted classifier to be utilized for the Supervised Learning module to detect the Concept Drift. Rust-Boosted classifier performed well in the Concept Drift scenario and maintained its performance accuracy better than other classifiers.

5. Conclusion

Through the comprehensive literature analysis on the CD detection techniques, we can safely state that still the concept drift detection is not deterministic, and several limitations can be highlighted from the proposed methods. For example, the difference between the actual CD and noise, the existing solutions cannot learn from the multiple concepts, the need a massive amount of data to be analyzed the drift pattern. Therefore, this study is a step towards the proposal of Concept Drift detection techniques to somehow minimize this limitation. Thus, in this study, we introduced a concept drift detection techniques. This technique utilizes the essence of both Supervised and Unsupervised Machine Learning approaches to find the potential Concept Drift. Initially, several datasets (SEA, IRIS, and Employee) dataset are investigated using the different configurations of K-Mean clustering. The propose of these experiments is to measure potential class boundaries without the proposal of Concept Drift detection. Our technique has the potential to become computationally efficient and straightforward to implement the Data

Distribution based Concept Drift Detection technique. In the initial experiments, we demonstrate empirically its effectiveness, not only for choosing the number of clusters but also for identifying underlying structure, on a wide range of newly created and available real-world datasets. Finally, we note that these ideas potentially can be extended towards defining the statistical approach for dynamical selection of the number of clusters in Unsupervised Learning problems.

Acknowledgments

This research study is a part of the funded project (A novel approach to mitigate the performance degradation in big data classification model) under a matching grant scheme supported by University Technology Petronas (UTP), Malaysia, and Hamdard University, Pakistan.

References

[1] Webb, Geoffrey I., et al. "Characterizing concept drift." Data Mining and Knowledge Discovery 30.4 (2016): 964-994.
[2] Jameel, Syed Muslim, et al. "A Fully Adaptive Image Classification Approach for Industrial Revolution 4.0." International Conference of Reliable Information and Communication Technology. Springer, Cham, 2018.
[3] Lu, Jie, et al. "Learning under concept drift: A review." IEEE Transactions on Knowledge and Data Engineering (2018).
[4] Shirkhorshidi, Ali Seyed, Saeed Aghabozorgi, and Teh Ying Wah. "A comparison study on similarity and dissimilarity measures in clustering continuous data." PloS one 10.12 (2015): e0144059.
[5] Mélisande Albert, Yann Bouret, Magalie Fromont, Patricia Reynaud-Bouret. Bootstrap and permutation tests of independence for point processes. Annals of Sta-tistics, Institute of Mathematical Statistics, 2015, 43 (6), pp.2537-2564. ff10.1214/15-AOS1351ff. ffhal-01001984v4f.
[6] Ikonomovska E., Gama J. and Dzeroski S. "Learning model trees from evolving data streams". Data Min. Knowl. Discov., 23, 1, 128-168, 2011.
[7] Mouss H., Mouss D., Mouss N. and Sefouhi L. "Test of Page-Hinckley, an ap-proach for fault detection in an agro-alimentary production system". In Proceedings of the Control Conference, 2004. 5th Asian. 815-818 Vol.812, 2004.
[8] Gama J., Medas P., Castillo G. and Rodrigues P. "Learning with Drift Detection". Springer Berlin Heidelberg,286-295, 2004.
[9] Gomes J. B., Menasalvas E. and Sousa P. A. C. "Learning recurring concepts from data streams with a context-aware ensemble". In Proceedings of the Proceedings of the 2011 ACM Symposium on Applied Computing TaiChung, Taiwan, ACM. 994-999, 2011.
[10] Bach S. H. and Maloof M. "Paired Learners for Concept Drift". In Proceedings of the Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on. 23-32, 2008.
[11] Bifet A. and Gavalda R. "Kalman filters and adaptive windows for learning in data streams". In Proceedings of the Proceedings of the 9th international conference on Discovery Science Barcelona, Spain, Springer-Verlag. 29-40, 2006.
[12] Harries M. B., Sammut C. and Horn K. "Extracting Hidden Context". Mach. Learn., 32, 2, 101-126, 1998.
[13] Klinkenberg R. "Learning drifting concepts: Example selection vs. example weighting". Intell. Data Anal., 8, 3, 281-300, 2004.
[14] Zliobaite I. "Learning under Concept Drift: an Overview". Computing Research Repository (CoRR), 1, 4784, 2010.
[15] Vorburger P. and Bernstein A. "Entropy-based Concept Shift Detection". In Pro-ceedings of the Data Mining, 2006. ICDM '06. Sixth International Conference on. 1113-1118, 2006.
[16] Baena-Garcıa, Manuel, et al. "Early drift detection method." Fourth internation-al workshop on knowledge discovery from data streams. Vol. 6. 2006.
[17] Gama J., Žliobaite I., Bifet A., Pechenizkiy M. and Bouchachia A. "A Survey on Concept Drift Adaptation". ACM Computing Surveys, 46, 4, 35, 2013.
[18] Bifet A. and Gavalda R. "Adaptive Learning from Evolving Data Streams". In Proceedings of the Proceedings of the 8th International Symposium on Intelligent Data Analysis: Advances in Intelligent Data Analysis VIII Lyon, France, Springer-Verlag. 249-260, 2009.
[19] Valli Udine, Syed Sajjad Hussain Rizvi, Manzoor Ahmed Hashmani, Syed Muslim Jameel, Tayyab Ansari. A Study of Deterioration in Classification Models in Re-al-Time Big Data Environment, the 4th International Conference of Reliable Infor-mation and Communication Technology 2019.

	Confusion Matrix			Tue Positive Rate	False Negative Rate	Training Accuracy (%)	Testing Accuracy (%)	Prediction Speed (Obs/sec)	Train Time (sec)
Tree	Fine Tree	Predict Class	0	97	3				
			1	14	86				
	Medium Tree	Predict Class	0	99	1	87.2	37.4033	~230000	1.2703
			1	4	96				

Appendix

	Complex Tree	Predict Class	0	100	0	87.4	37.3067	~270000	0.88521
			1	0	100				
Discriminant Analysis	Linear Discriminant	Predict Class	0	95	5	86.3	53.7033	~430000	1.2803
			1	28	72				
	QuadraticDiscriminant	Predict Class	0	95	5	86.3	54.1000	~1100000	1.002
			1	28	72				
Support Vector Machine	Linear SVM	Predict Class	0	100	0	87.4	37.3067	~43000	18.159
			1	0	100				
	Quadratic SVM	Predict Class	0	100	0	87.4	54.2067	~27000	1110.5
			1	0	100				
	Cubic SVM	Predict Class	0	79	21	75.7	58.9183	~23000	1475.5
			1	53	47				
	Fine Gaussian SVM	Predict Class	0	98	2	86.9	42.5583	~11000	72.023
			1	12	88				
	Medium Gaussian SVM	Predict Class	0	100	0	87.4	37.3067	~11000	74.932
			1	0	100				
	Coarse Gaussian	Predict Class	0	100	0	87.4	62.9650	~15000	87.884
			1	0	100				

Table. 1: The performance of Shallow Learning Classification Models using SEA dataset.

	Confusion Matrix			Tue Positive Rate	False Negative Rate	Training Accuracy (%)	Testing Accuracy (%)	Prediction Speed (Obs/sec)	Train Time (sec)
KNN	Fine KNN	Predict Class	0	91	9	83.6	62.9650	~100000	7.053
			1	35	65				
	Medium KNN	Predict Class	0	97	3	86.6	46.9583	~54000	7.2397
			1	15	85				
	Coarse KNN	Predict Class	0	>99	<1	87.3	38.0083	~21000	9.477
			1	1	99				
	Cosine KNN	Predict Class	0	97	3	86.8	45.3233	~7000	19.488
			1	13	87				
	Cubic KNN	Predict Class	0	97	3	86.6	46.7800	~26000	10.372
			1	15	85				
	Weighte	Pred.	0	94	6	85.5	58.4833	~79000	8.3205

Table 2: The performance of Shallow Learning Classification

			1	27	73				
Boosted Tree	Boosted Tree	Predict Class	0	98	2	86.8	39.6550	~68000	9.8194
			1	10	90				
Bagged Tree	Bagged Tree	Predict Class	0	95	5	86.0	56.2817	~55000	12.634
			1	25	75				
Subspace Discrim.	Subspace Discriminant	Predict Class	0	98	2	87.0	42.4250	~58000	6.7217
			1	9	91				
Subspace KNN	Subspace KNN	Predict Class	0	96	4	85.8	52.7650	~14000	10.06
			1	18	82				
RUSBoosted Tree	RUSBoosted Tree	Predict Class	0	79	21	79.9	84.9433	~110000	10.931
			1	87	13				