Journal of Information Systems Engineering and Management
2025, 10(9s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Data Stream Learning Evaluation: Experimenting with Prequential Approach Over Real Data Streams

Ms. Shailaja B. Jadhav [1], Dr. D. V. Kodavade [2], Ms. Vinaya D. Kulkarni[3]
[1]Asst. Prof. Dept. of Computer Engg. MMCOE, Pune, India
[1] Ph.D Research Scholar , Shivaji University, Kolhapur, Maharashtra, India
[2] Professor, DKTES' Textile and Engineering Institute, Ichalkaranji, Kolhapur. Maharashtra, India
[3] Asst. Prof. Dept. of Computer Engg. , BV'S College of Engg. for women, Pune, India
1 msgshalom@gmail.com, 2 dvkodavade@gmail.com, 3 vinaya.kulkarni@bhartividyapeeth.edu

ARTICLE INFO

Received: 27 Oct 2024

Revised: 20 Dec 2024

Accepted: 05 Jan 2025

ABSTRACT

Current IoT and sensor device-led computing systems are giving rise to enormous streaming data . This ever-growing streaming data necessitates data analytics to do ; on-the-fly decision making thus giving rise to stream learning enabled decisions. These new demands of data stream learning algorithms require evolving algorithms , evaluation methods and dynamic learning capabilities from the machine learning systems. Thus, changing data distribution over stream learning environments is very less suitable to do with the existent and static environments of data analytics. This article addresses the prequential learning evaluation in data stream learning environment over real data streams like covertypeand airline with scikit multiflow stream learning APIs . The major objective of the paper is to analyse the better alternative to traditional hold-out evaluation . The classifier machine learning algorithms like tree based adaptive classifiers and some ensemble-based alternatives are empirically evaluated over both hold-out and prequential methods. The results of the experimental evaluation suggest that prequential evaluation is able to achieve improvements in accuracy. Further this article has evaluated time complexity which again suggests that GPU and similar computing environments if utilized , prequential evaluation may be the optimal solution for dynamic stream learning environments.

Keywords: Data Analytics, Stream learning, Classification, Prequential evaluation, Adaptive classifiers.

1. INTRODUCTION

Stream learning has attracted many researchers in recent years because practically every program used on a daily basis generates an enormous amount of data. Stream learning is defined as a continuous, infinite input of observations, with the most recent ones being the most significant. Characteristics are altering dramatically and in an almost unforeseen manner over time due to the very dynamic nature of the data being inserted [10]. Furthermore, it is impractical to keep all historical samples due to the massive volume of data, making it difficult to extract real-time knowledge. [10]

2. REAL TIME DATA STREAM LEARNING

The method by which the data is made available to the system is a crucial and fundamental component of streaming data. High latency data sources will "hold" the entire system, making learning algorithms challenging [1]. Beginners frequently find it more difficult to understand the data source for streaming data because of the significant distinctions, such as the fact that it is not only a self-contained file or a clearly defined database. To ensure that the learner is updated as soon as new data becomes available, it must really permit the presentation of new data with minimal delay. Many frameworks for stream processing, like Apache Spark, Flink, and SKmultiflow, manage streaming data, addressing its dynamic, ever-changing behavioural patterns and attempting to extract knowledge instantly.

3. RELATED WORK

In offline classification [6], the classifier algorithm is provided a collection of labelled cases. It builds a model and trains it on labelled instances, such that when an unlabelled instance arrives, it can predict the label for the unseen instances. In batch classification, the dataset is divided into a training set and a test set, or an alternate method is cross-validation, to ensure that the classifier is reasonably accurate. In all cases, the training and prediction phases must be separated by time. In streaming or online classification, the distinction between training and testing becomes ambiguous and intertwined. The prediction begins before all of the data arrives, as incoming data may never end.Thereare many wide research efforts done in this regard [3,4,6], and after a thorough studies most prominent ones are highlighting use of decision tree alternatives and ensemble based classifiers to name a few.

Because of the simplicity of their design, decision tree learners are the most popular for stream categorization [11,13]. Decision tree classifiers are popular because (a) they do not require domain knowledge, (b) they handle high-dimensional data, (c) they are simple and fast, (d) they produce accurate results, and (e) the predictions are relatively easy to grasp. It supports both discrete and continuous-valued characteristics. Traditional decision trees cannot be applied directly to streaming data because they increase the cost of time and memory use. Instead of looking at previously stored instances to select the splitting criterion, the Hoeffding tree waits until enough examples are received to determine the splitting criterion attribute.The key advantage of the Hoeffding tree is that it does not require storing prior instances. The ability to split property can be determined only by mathematical computation.One more popular approach for classifying data streams is ensemble-based methods. Because they produce better results than depending on strong single learners, they are well-liked by researchers [4,5,8]. The relative ease of implementation in practical applications is another attribute of ensemble-based classification [5]. Because ensemble methods may be used with drift detection algorithms and contain dynamic updates, like the addition of classifiers or the selective removal of others, they are particularly helpful for data stream learning.

Ensemble classifiers use a variety of data stream classification techniques, many of which are meta-algorithms. This involves combining numerous single classifiers using different heuristic approaches [6]. There isn't a general guideline for creating the ensemble that yields the optimal solution for each problem [6,7]. To address various issues, different combination techniques are applied to a range of fundamental algorithms. Ensemble learning aims to generate a combinatorial approach with multiple models and depends on their vote for final predictions, in contrast to incremental learning, which aims to give a single model [4]. Because the model is trained on a tiny sample of stream data, it may be utilized to effectively handle rapidly increasing data quantities, making this architecture beneficial for handling data streams. For data stream analysis, the ensemble-based approach—where the choice made by the group of algorithms can be viewed as a decision based on the opinions of certain expertsis more widely used [4].

The classifier does not have access to learning instances in real-time streaming data scenarios like it would in a static learning environment; instead, they are supplied sequentially and with a high instream velocity. Data stream classifiers should therefore be able to learn every instance in a single scan or, if required, store them for a brief amount of time. Ensemble learners respond strongly to this data stream-specific task.

It is much obvious here, the way in which data stream classification needs to be evaluated. Data stream classification mostly revolves around two methods of evaluation[16] hold-out and prequential. In hold-out evaluation the test and train sets are predefined. It can measure the accuracy very precisely for the current instances. It is a less suitable method streaming instances. Moreover, it becomes challenging to obtain recent-most significant test data for the stream learning. Hence, more accurate method of evaluation i.e. Prequential or Interleaved Test-Then-Train is now experimented with many of the stream learning researches [16], hereeach separate instance is used to test the model before the instance is used to train the model. In the prequential evaluation method, the accuracy can be updated incrementally.

4. RESEARCH OBJECTIVE

- To find out the suitable alternative to holdout evaluation .
- To choose from standard machine learning tree based and ensemble classifiers for streaming data analytics with prequential approach.

5. PREQUENTIAL EVALUATION

One common estimation method in dynamic contexts is the prequential approach. There is no need for a distinct holdout set because, as the image illustrates, all incoming instances are used for training and testing. This method is also known as the Interleaved Test-Then-Train approach, which maximizes the use of the available data by evaluating the model's accuracy on each individual instance and updating it incrementally as new instances are added to the stream. Strictly following this sequence reduces the significance of each individual case to the average.[9]

In dynamic learning scenarios, the prequential approach is a majorly applied choice of estimation procedure. The distinguishing merit of this prequential approach comes from training phase incorporated here. There is no need for distinct holdout set, instead of that, all the instances are used in training and testing , thus making maximum use of large and varying nature of incoming data instances, which makes it the most suitable choice for streaming data analysis.

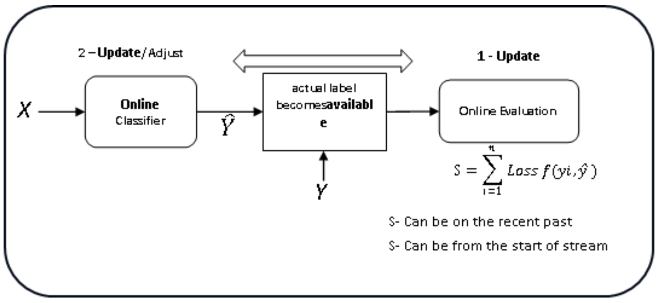

2 – Update/Adjust
1 - Update
X
Online Classifier
$\hat{Y}$
actual label becomes available
Y
Online Evaluation
$S = \sum_{i=1}^{n} Loss\, f(yi, \hat{y})$
S- Can be on the recent past
S- Can be from the start of stream

Fig.1 Pictorial representation of Prequential evaluation

Prequential approach for evaluation is actually predictive sequential (prequential). To be simpler and more generic, it can be understood as a learning mechanism where each incoming instance from a data stream can be used to test the classifier (any machine learning model) and does not follow the traditional approach of training the machine learning model over some dedicated train block of data and then apply it over a separate test data. As given by A. Bifet et al in [9] and Hidalgo et al [16] with prequential approach as evaluation measure for data stream learning algorithm, the model is being tested over the data stream instance before it is used to train and incrementally update the evaluation performance over the entire data stream, order matters here, first test and then train, i.e. interleaved test then train. Thus, the model is being tested over the unseen examples and the accuracy over the current testing is continuously updated in incremental manner. As time passes on, each individual instance becomes less contributable and not much significant, it really captures the dynamic nature of a real time data stream. Thus, one can get smooth plot of overall average accuracy over time, utilising the available data to the fullest possible potential. Eq. (1) shows the summative S calculation for prequential approach.

$$S = \sum_{i=1}^{n} lossf(y, \hat{y}) \qquad (1)$$

The prequential accuracy is calculated online as shown in the Eq. (2). The prequential accuracy can be calculated from the beginning of the stream over a sliding window of the fixed size or can be over instances seen in recent past(fading factor) . Using these variations sliding window and fading factor , it is better able to represent the average accuracy . In other words, it is the accuracy given by each example'sprediction before it is learned. At timestamp t , the prequential accuracy p_acc(t) is can be represented asEq. 2

$$p_{\_acc(t)} = acc\_ex(t), \qquad if\ t = f$$

$$p_{\_acc(t)} = P\_acc(t-1) + \frac{acc\_ex(t) - P\_acc(t-1)}{t-f+1}, \qquad otherwise(2)$$

Where, acc_*ex* is 1 if correctly classified current example and 0 otherwise, f denotes every calculation's first timestamp, i.e, the timestamp when each concept drift is detected.

The results of the carried-out experiments suggest that the use of prequential evaluation with the sliding window variation is the best alternative [16] to learn in dynamic learning environments as covertype and airline data w.r.t. the underlined research.Following algorithm shows a pseudocode for the prequential approach.

5.1 Pseudo-algorithmic steps for prequential evaluation

```
Step 1.  While no. of records <max. recordsand
Stream.has_more samples
Step 2.  X, Y=stream.next_sample()//Get one instance from incoming stream
//without clearly dividing as train test or specific hold-out set
Step 3.  Y_pred=Classifier. Predict(X)
//test the instance
Step 4   if Y[0]==Y_pred[0]:  //check performance-Predict label for new data
Step 5.  Add the current classifier to best performing classifiers
         And Model will be trained Incrementally with the new data
Step 6.  Take the next record and repeat //Interleaved Test-then -train
```

Thus, more suitable to dynamic learning scenarios, the majorly applied estimation procedure is prequential evaluation . All the streaming instances are used in training and testing , so, it helps to gain insights from varying nature of incoming data and making use of large speedydata to the fullest potential.

6. EXPERIMENTAL WORK

The Scikit Multiflow architecture for stream learning is used for all of the experiments described here. The present experimentation has used Covertype dataset rom UCI Machine Learning repository. It is openly available and most used for stream data analysis tasks. Scikit multi-flow framework has a wide variety of classes and APIs for data stream prequential learning environments. Table 1 shows the experimental setup details and programming environment. Readers are requested to take a note here that , GPU computation is recommended here, authors have planned to take it in the further research.

Table 1 Experimental setup

Sr.	Parameters	Values
1	Framework	Scikit Multiflow
2	Dataset	Cover type –[19],Airlines -Kaggle [18]
4	CPU	Intel Core i7 16 GB RAM
6	Programming	Python 3.6.9, scikit learn
8	Anaconda Jupyter	Version 5.3.1, Version 5.7.2 ; x86_64 bit

7. RESULTS AND DISCUSSIONS

The performance of the machine learning methods is assessed using cover type and airline data, as shown in Table 2. This experimentation is conducted using the Skmultiflow platform [21]. For data stream classification task in particular, each classifier is assessed across small and large sample space. This analysis uses accuracy and time as evaluation metrics. Among the classifier algorithms, the Hoeffding Tree Classifier, Hoeffoding Adaptive Tree classifiers, and Adaptive Random Forest classifier have demonstrated improvements in accuracy, with prequential evaluation , while ensemble variations like dynamic weighted majority (DWM),accuracy updated ensemble (AUE) and accuracy enhanced ensemble (AEE) have also demonstrated a notable improvement in accuracy in covertype data , although results for airline data are not so prominent.

For data stream analysis , the difference in performance decreases with the number of instances at a broad scale. Further significant fact is that these algorithms allow for a substantial increase over time, despite their time complexity(this necessitates GPU computation environment). The algorithm's most admirable feature for real-time data stream analysis is its ability to tune with the incoming data instances viz. Hoeffding adaptive tree (HAT), achieve a remarkable level of accuracy in a short length of time. ARF is time consuming as initially it takes much train time, but on the increasing timeline test time taken is lesser . (Refer Table 2 and 3)

For interested readers, in detail evaluation graphs with each of the stream data are shown in annexure A and annexure B. Where, Annexure A part- I shows the evaluation graphs for Covertype and part- II shows it for Airlines stream data. In order to analyse individual time taken by each of the tree based and ensemble algorithms , Annexure B is there , showing evaluation graphs for prequential analysis , with covertype data, depicting accuracy as well as time taken(train time, predict time, total).

Table 2 Accuracy Analysis

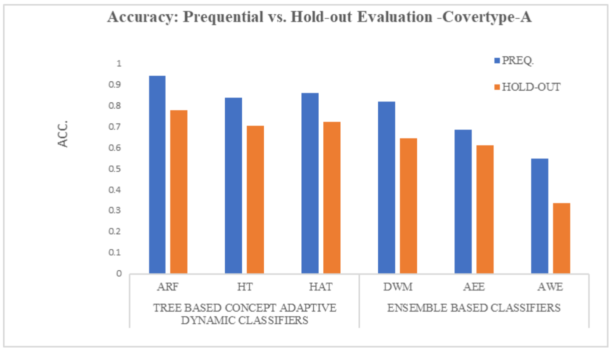
Covertype A -0.2m	**Accuracy**					
Evaluation Type	**Tree based adaptive classifiers**			**Ensemble classifiers**		
	ARF	**HT**	**HAT**	**DWM**	**AEE**	**AWE**
Prequential	0.9398	0.8372	0.8582	0.8193	0.6867	0.547
Hold-out	0.7782	0.7041	0.721	0.6452	0.6126	0.3369
Covertype B -50k	**Accuracy**					
Evaluation Type	Tree based adaptive classifiers			Ensemble classifiers		
	ARF	HT	HAT	DWM	AEE	AWE
Prequential	0.9106	0.8331	0.8135	0.7909	0.6939	0.5027
Hold-out	0.7295	0.7113	0.7615	0.7063	0.6986	0.407
Airlines – (0.1m)	**Accuracy**					
Evaluation Type	Tree based adaptive classifiers			Ensemble classifiers		
	ARF	HT	HAT	DWM	AEE	AWE

Prequential	0.643	0.6508	0.6494	0.6159	0.6519	0.5379
Hold-out	0.6597	0.6596	0.6626	0.6089	0.6624	0.5864
Airlines -50k	**Accuracy**					
Evaluation Type	Tree based adaptive classifiers			Ensemble classifiers		
	ARF	HT	HAT	DWM	AEE	AWE
Prequential	0.6365	0.6473	0.6429	0.6082	0.6395	0.6293
Hold-out	0.6217	0.6147	0.6139	0.6118	0.6131	0.6127

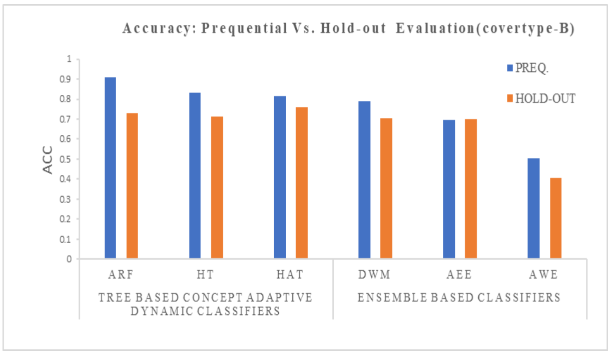
Table3 Time Analysis

Covertype A-50k	**Time (S.)**	
Evaluation Type	**Tree based adaptive classifiers**	**Ensemble classifiers**
Prequential	565.84	1693.14
Hold-out	1228.56	912.87
Airlines -50k	**Time (S.)**	
Evaluation Type	**Tree based adaptive classifiers**	**Ensemble classifiers**
Prequential	528.75	328.46
Hold-out	330.97	157.9

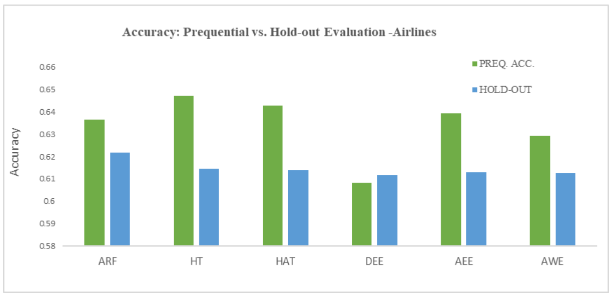
Fig. 2 a-c shows the evaluation graphs of the conducted experiments.



(a)



(b)



(c)

Fig. 2(a-c) – Graphs showing prequential vs. hold-out evaluation a) Covertype-A b) Covertype-B c) Airlines

8. CONCLUSION AND FURTHER RESEARCH

This work focuses on data stream learning with main objective to adapt the existing machine learning classifiers using a prequential evaluation approach. Experiments have been conducted on the representative scales of two data streams covertype and airlines. Although, the present research is carried out without incorporating GPU environment, the results often demand the extensive need of high computing power(especially time analysis for stream learning). One of the major observations through the experiments is that combining a prequential learning approach with the traditional test-then-train methodology may yield better results. The HT and HAT classifier performs consistently well in all the data streams in prequential evlaution approach. ARF performs suitably , but requires large train time. Ensemble classifier's results suggest that there is no notable improvement in prequential evaluation, and it is much needed to tune ensemble classifiers for prequential evaluation. Also, time required for prequential evaluation is much larger but definitely can be tuned in well with GPU or similar extensive computing environments.

The assessment of machine learning classifiers can be further improved by scikitmultiflow data stream learning environment's hyperparameter tuning feature, especially in the data stream learning environment [9,10]. The researchers in this field can extend their experimental and research efforts to do more sophisticated analytic procedures that incorporate other assessment criteria such as computation time, memory, degree of parallelism (if applicable), etc. [13,14], the present work has limitations of computing power , so could not incorporate. The other datasets can also be experimented with using prequential analysis to find additional research directions .

REFERENCE:

[1] Heitor Murilo Gomes, Jean Paul Barddal, and Fabr´icio enembreck , " A Survey on Ensemble Learning for Data Stream Classification", ACM Computing Surveys, Vol. 50, No. 2, March 2017

[2] Ms. Shailaja B. Jadhav, Dr. D. V. Kodavade, 'A Comprehensive –Theoretical Review of Data Stream Mining:methods and challenges', IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 23, Issue 3, Ser. II (May – June 2021), PP 55-67

[3] Leszek Rutkowski, Maciej Jaworski, Piotr Duda, "Stream Data Mining: Algorithms and Their Probabilistic Properties " ISBN 978-3-030-13961-2 eBook

[4] Van Rijn, J.N., Holmes, G., Pfahringer, B. et al. The online performance estimation framework: heterogeneous ensemble learning for data streams. Mach Learn 107, 149–176 (2018).

[5] Ms. Shailaja B. Jadhav, Dr. D. V. Kodavade, ' Performance Analysis of Ensemble Learning For Artificial And Real Time Data Streams - Research Directions', International Conference on Recent Advances in Industry 4.0 Technologies , Sept. 22- NIT Pudducherry, India.

[6] Bartosz Krawczyk, Leandro L. Minku, João Gama, Jerzy Stefanowski, MichałWoźniak, "Ensemble learning for data stream analysis: A survey", Information Fusion, Volume 37,2017

[7] Dariusz Brzezinski, Jerzy Stefanowski, "Combining block-based and online methods in learning ensembles from concept drifting data streams", Information Sciences, Volume 265, 2014, Pages

[8] Peng Zhang, Xingquan Zhu, Yong Shi, Li Guo, Xindong Wu, "Robust ensemble learning for mining noisy data streams", Decision Support Systems , Volume 50, Issue 2,2011, Pages 469-479

[9] Albert Bifet. Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams. IOS Press, 2010.

[10] Gomes, H. M., Read, J., Bifet, A., Barddal, J. P., & Gama, J. (2019). Machine learning for streaming data: state of the art, challenges, and opportunities. ACM SIGKDD Explorations Newsletter, 21(2), 6-22.

[11] Zhang, H., Liu, W., & Liu, Q. (2020). Reinforcement Online Active Learning Ensemble for Drifting Imbalanced Data Streams. IEEE Transactions on Knowledge and Data Engineering.

[12] Zhao, Y., Wang, S., Wang, Y., & Liu, H. (2020). Stratified and Time-aware Sampling based Adaptive Ensemble Learning for Streaming Recommendations.

[13] Srirutchataboon, G., Prasertthum, S., Chuangsuwanich, E., Pratanwanich, P. N., & Ratanamahatana, C. (2021). Stacking Ensemble Learning for Housing Price Prediction: A Case Study in Thailand. KST 2021 - 2021 13th International Conference Knowledge and Smart Technology, 73–77.

[14] Sun, Y., & Dai, H. (2021). Constructing accuracy and diversity ensemble using Pareto-based multi-objective learning for evolving data streams. Neural Computing and Applications, 33(11),

[15] Li, J., Hao, J., Feng, Q. Q., Sun, X., & Liu, M. (2021). Optimal selection of heterogeneous ensemble strategies of time series forecasting with multi-objective programming. Expert Systems with Applications, 166. https://doi.org/10.1016/j.eswa.2020.114091

[16] Gonzalez Hidalgo, Juan & Maciel, Bruno Iran Ferreira & Barros, Roberto. (2019). Experimenting with prequential variations for data stream learning evaluation. Computational Intelligence. 35.

[17] The OpenSky Network,https://opensky-network.org

[18] airline data- https://www.kaggle.com/datasets/bulter22/airline-data

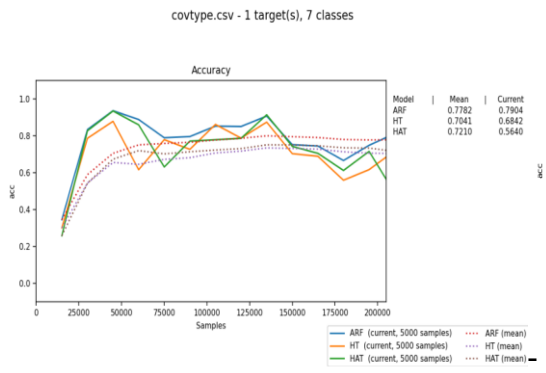
[19] covertype data – https://archive.ics.uci.edu/dataset/31/covertype

[20] Jadhav, S. B. ., & Kodavade, D. V. . (2023). Enhancing Flight Delay Prediction through Feature Engineering in Machine Learning Classifiers: A Real Time Data Streams Case Study. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2s), 212–218.

[21] Montiel, J., Read, J., Bifet, A., & Abdessalem, T. (2018). Scikit-multiflow: A multi-output streaming framework. The Journal of Machine Learning Research, 19(72):1–5.

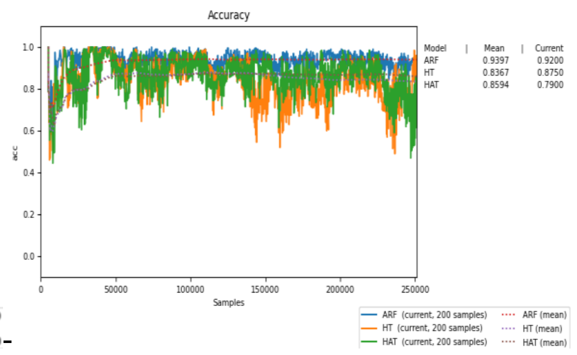
Annexure A –Part-I- Cover type stream Data

Evaluation graphs for tree based and ensemble classifiers, hold-out and prequential approach



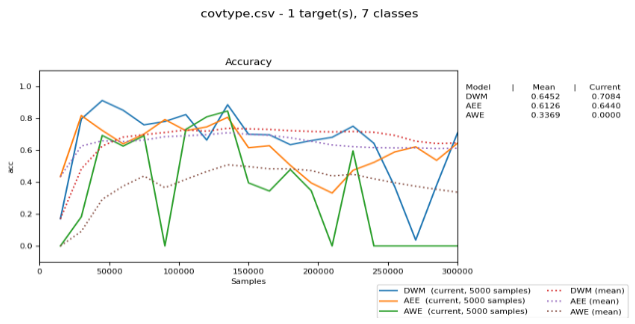
Model	Mean	Current
ARF	0.7782	0.7904
HT	0.7041	0.6842
HAT	0.7210	0.5640

(a)



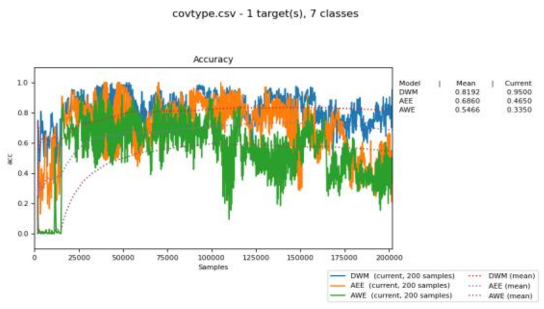
Model	Mean	Current
ARF	0.9397	0.9200
HT	0.8367	0.8750
HAT	0.8594	0.7900

(b)



Model	Mean	Current
DWM	0.6452	0.7084
AEE	0.6126	0.6440
AWE	0.3369	0.0000

(c)



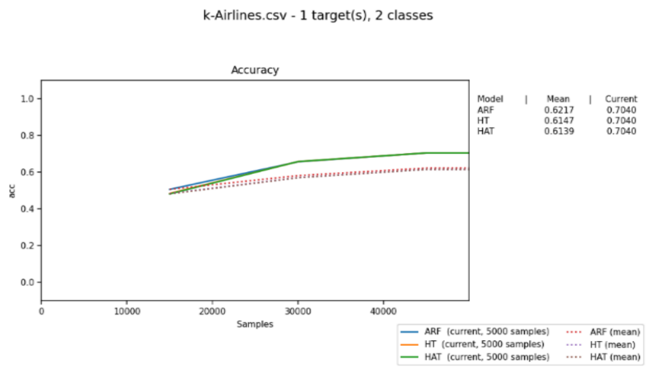
Model	Mean	Current
DWM	0.8192	0.9500
AEE	0.6860	0.4650
AWE	0.5466	0.3350

(d)

Fig. Evaluation Graphs a) hold-out Tree based classifiers b) Prequential – Tree based classifiers c) holdout- ensemble based classifiers d) Prequential - ensemble based

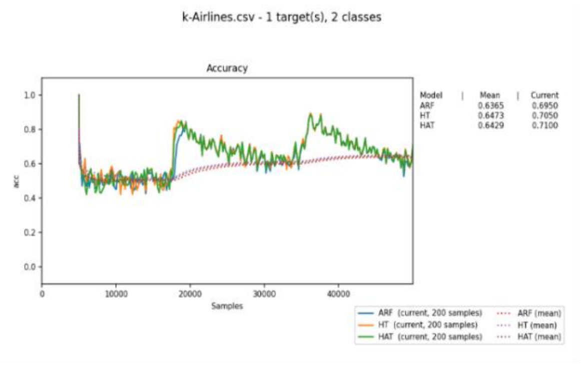
Annexure A-Part-II – Airlines stream Data

Evaluation graphs for tree based and ensemble classifiers, hold-out and prequential approach



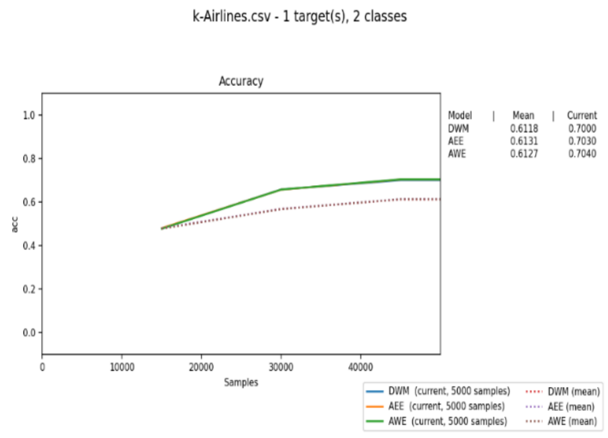
Model	Mean	Current
ARF	0.6217	0.7040
HT	0.6147	0.7040
HAT	0.6139	0.7040

(a)

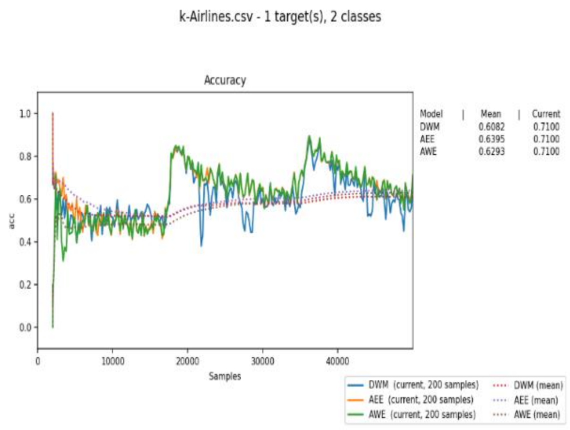


Model	Mean	Current
ARF	0.6365	0.6950
HT	0.6473	0.7050
HAT	0.6429	0.7100

(b)



Model	Mean	Current
DWM	0.6118	0.7000
AEE	0.6131	0.7030
AWE	0.6127	0.7040

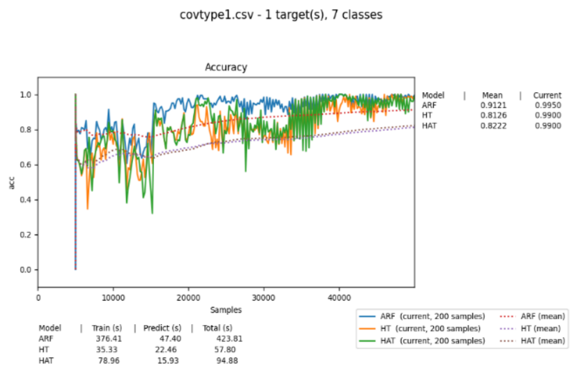


Model	Mean	Current
DWM	0.6082	0.7100
AEE	0.6395	0.7100
AWE	0.6293	0.7100

(c) **(d)**

Fig. Evaluation Graphs a)Hold-out- Tree based classifiers b) Prequential Tree based classifiers c) Hold-out Ensemble based classifiers d) Prequential – Ensemble Based Classifiers

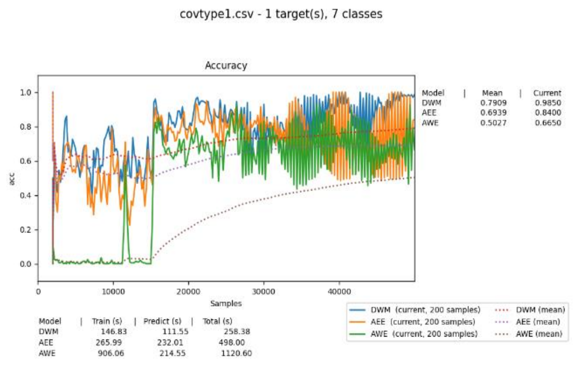
Annexure B – Detailed illustrative graphs -sample representation - Prequential evaluation Covertype B-with each classifier (*Accuracy And Time*)



Model	Mean	Current
ARF	0.9121	0.9950
HT	0.8126	0.9900
HAT	0.8222	0.9900

Model	Train (s)	Predict (s)	Total (s)
ARF	376.41	47.40	423.81
HT	35.33	22.46	57.80
HAT	78.96	15.93	94.88

(a) Prequential Evaluation – Tree based classifiers



Model	Mean	Current
DWM	0.7909	0.9850
AEE	0.6939	0.8400
AWE	0.5027	0.6650

Model	Train (s)	Predict (s)	Total (s)
DWM	146.83	111.55	258.38
AEE	265.99	232.01	498.00
AWE	906.06	214.55	1120.60

(b) Prequential Evaluation – Ensemble based classifiers