Custom Ensemble Model Implementation for Concept Drift Handling and Comparison with State of Art Models regarding Classification Task

Eray Gündoğdu

Department of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Turkey

Introduction

Occasionally, the straightforward batch training approach in machine learning is not the sufficient to obtain a high acccuracy classifier. To be more specific in real life scenarios the underlying distribution in data may change hence the model should be adapted to such kind of changes. This report is dedicated to investigate the concept drift in data regarding the classification task. Throughout the assignment two synthetic and two real dataset with concept drifts are used. These datasets are fed to the two state of art models which are Adaptive Random Forest (ARF) and Streaming Agnostic Model with k-Nearest Neighbors (SAM-kNN) and the evaluation metrics which are specified as overall accuracy and prequential accuracy plot are observed. As the next step a custom ensemble model is implemented with active drift detection by using DDM, EDDM, and ADWIN drift detectors. As the following step a custom ensemble model with passive drift detection is implemented. The influence of parameters and comparison of passive drift detection model, active drift detection model and, state of art models is provide later in the report in the context of evaluation criteria in Section 5.

Dataset Generation

As mentioned in the introduction througout the implementations 4 datasets are used. Two of them are synthetic datasets and two of them are real datasets. Real datasets are Spam and electricity datasets in a csv file which are extracted form the provided github link. The synthetic datasets are generated using AGRAWALGenerator and SEADataset. Both of the synthetic data has 100.000 samples which have abrupt drift with width=1 at the 35k and 60k positions. These generated datasets are saved in a csv file for latter use.

4. Classification Task with Concept Drift Handling

4.1

Before the implementation of custom ensemble models, state of art classfication models for concept drift detection. The models are ARF and SAM-kNN. The models are tested on 4 datasets, 4 prequential accuracy plots are obtained and the overall accuracies of the models are provided on the legend of the plots. The number of base learnes in ARF is fixed to 10 and the number of neighbors in SAM-Knn is fixed to 8. It is benefitial to mention that the models are warmed up (trained) with 2000 samples afterwards the prequential accuracy plot is generated by keeping the accuracy in window sizes which is determined by the number of samples divided by 20 to have in total 20 windows.

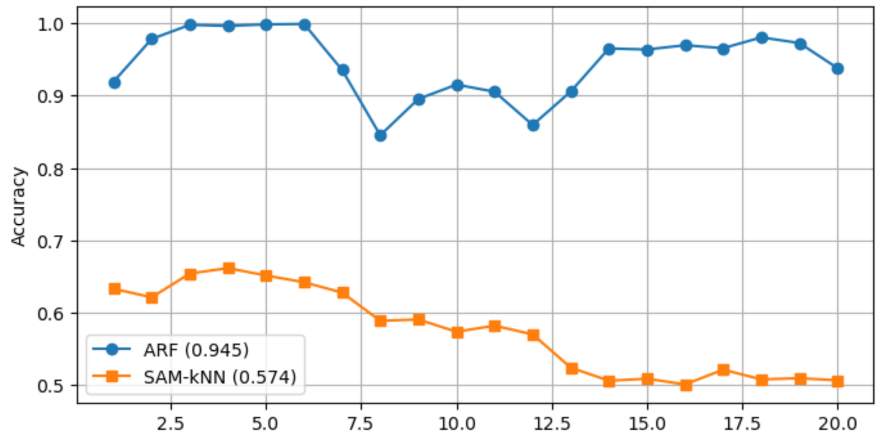

Figure 1 Prequential Accuracy as a function of window number on AGRAWALGenerator.csv

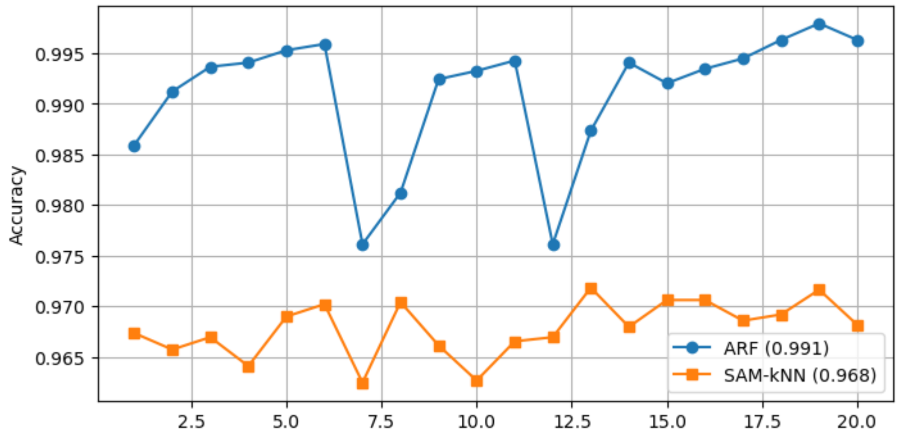

Figure 2 Prequential Accuracy as a function of window number on SEADataset.csv

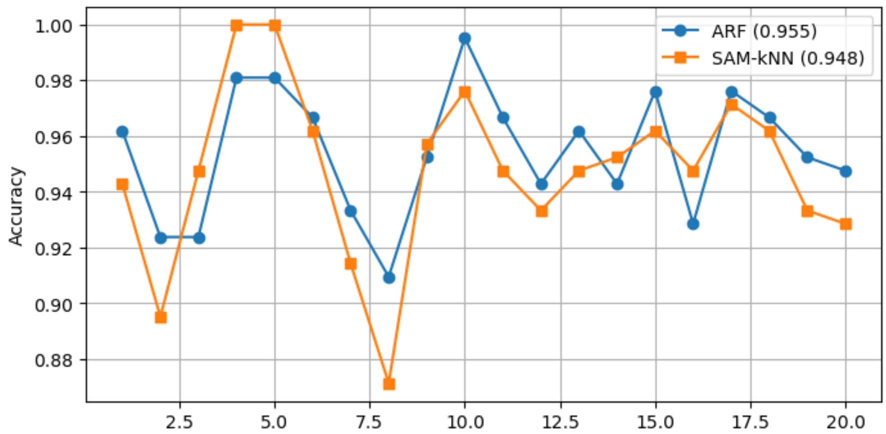

Figure 3 Prequential Accuracy as a function of window number on spam.csv

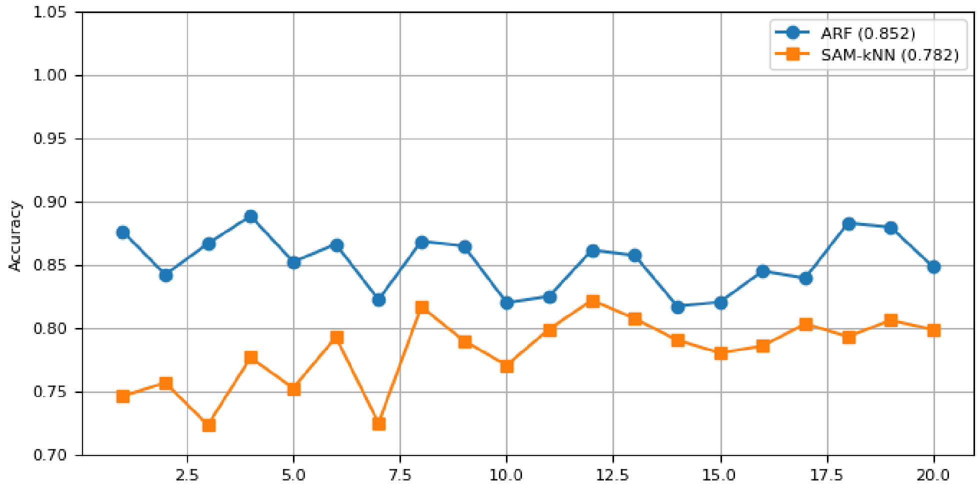

Figure 4 Prequential Accuracy as a function of window number on elec.csv

After the observations the concept drift in synthetically generated datasets can be verified due to accuracy drop around window number 7 and 12. Also the behaviour of real life datasets can beobserved. The poor accuracy performance of SAM-Knn on elec.csv and AGRAWALGenerator.csv is remained unsolved.

4.2

In this section an custom ensemble model is implemented using HoeffdingClassifier models as the base learners. The aim is to implement a model such that it can handle the concept drift in all of the datasets. It is important to mention that in this section the ensemble model utilizes active drift detection meaning that to each of the base learners a drift detector (DDM, EDDM, ADWIN) is assigned. The assignment of drift detectors to base learners is performed in a round-robin manner. In addition as the adaptive weighting algorithm, weights of the base learners are assigned proportionally with the accuracy over the last window (influence of this window size is discussed in section 4.3).

Test then train logic in the implementation is as following, initially every base learner makes prediction, result in terms of binary error is fed to the detector, if detected.change() method return true the base learner is replaced with an untrained base learner and as the last step learner.partial_fit (custom function belongs to the class) is called to train the learner on the instance. The number of base learners is fixed to 5 by considering computational efficiency (also the accuracy does not increase remarkable after 5 base learners). Initially the model is warmed up with the help of training on 2000 samples. Later on the previously explained flow continues and prequential accuracies are obtained over 20 sliding windows.

Also, the reported accuracy over the window, in other words window size has no significant contribution in this section due to presence of strong active drift detectors. This statement can also be verified from the code. For this section the detected.change() determines the replacement of the learner wheras in 4.3 the average accuracy over a specific window size and the threshold value determines replacement of the learner. The influence of window size is mentioned in section 4.3. From the following figures overall accuracies are provided on the legend. The overall accuracies are 0.917, 0.976, 0.914, 0.848 for ARGAWALGenerator.csv, SEADataset.csv, spam.csv and elec.csv respectively.

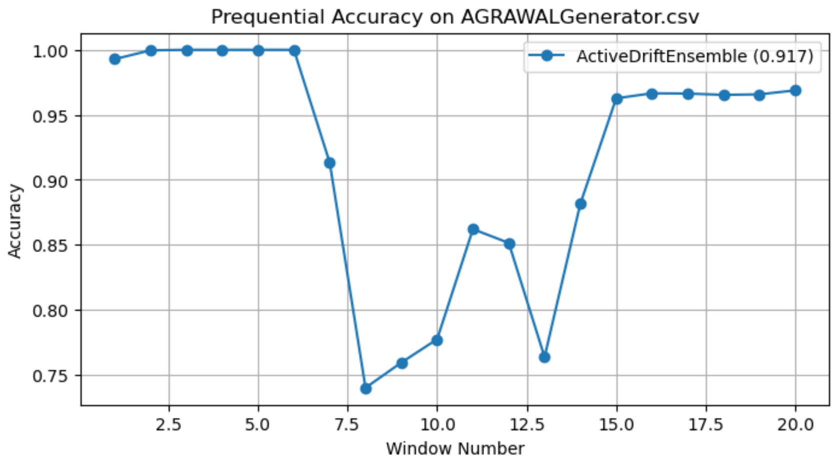

Figure 5 Prequential Accuracy as a function of window number on AGRAWALGenerator.csv

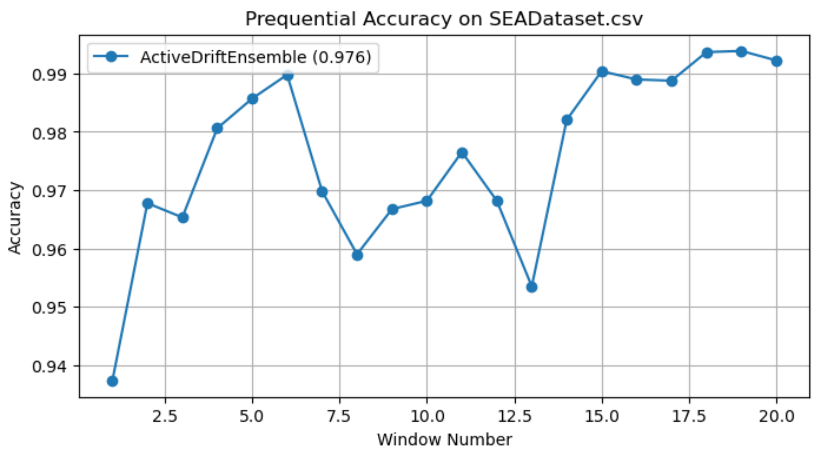

Figure 6 Prequential Accuracy as a function of window number on SEADataset.csv

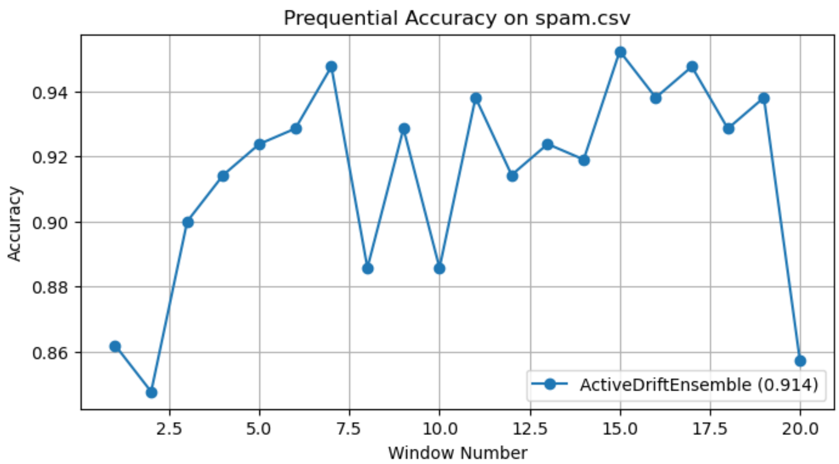

Figure 7 Prequential Accuracy as a function of window number on spam.csv

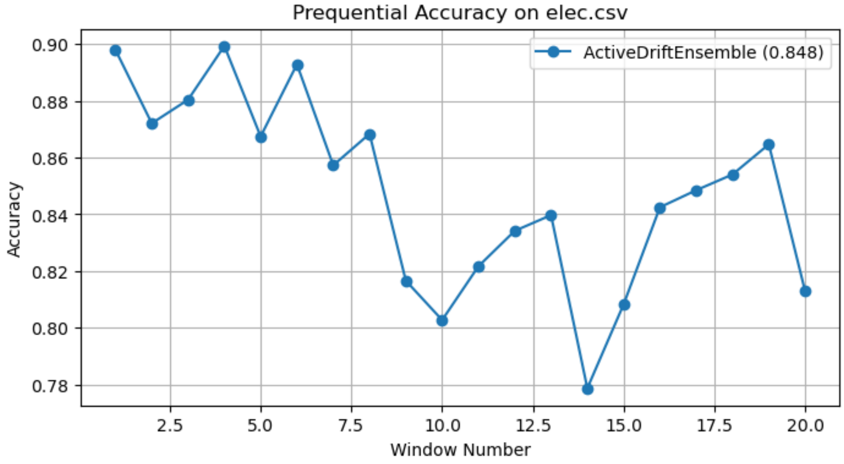

Figure 8 Prequential Accuracy as a function of window number on elec.csv

4.3

In this section the previous model is extented to be an ensemble model with passive drift detection. In this implementation the previous test then train and accuracy calculation with the window size is kept same. The changed part of the implementation is that, the replacement procedure of the base learner. In active drift detection case, the bool value returned from the detected.change() method determined the replacement of the base learner. In this section the replacement of the base learner is determined by numerically comparing the calculated accuracy over a window to a defined threshold. If the accuracy is less then the determiend threshold the base learner is replaced.

In this model the number of base learners is fixed to 5 and the window size is fixed to 150 which are the same with section 4.2. The influence of window size is provided in section 5. The additional parameter in this model is the threshold. Threshold determines the replacement of a base learner. As the threshold becomes larger the model produces less noise however it trades off with the model adaptability as the threshold increase the learners are not replaced and model can not adapt to concept drifts. This effect can be observed in Figure 9. The overall accuracy for 0.9 threshold is 0.6491, the overall accuracy for 0.50 threshold is 0.951. Hence the threshold is fixed to 0.50.

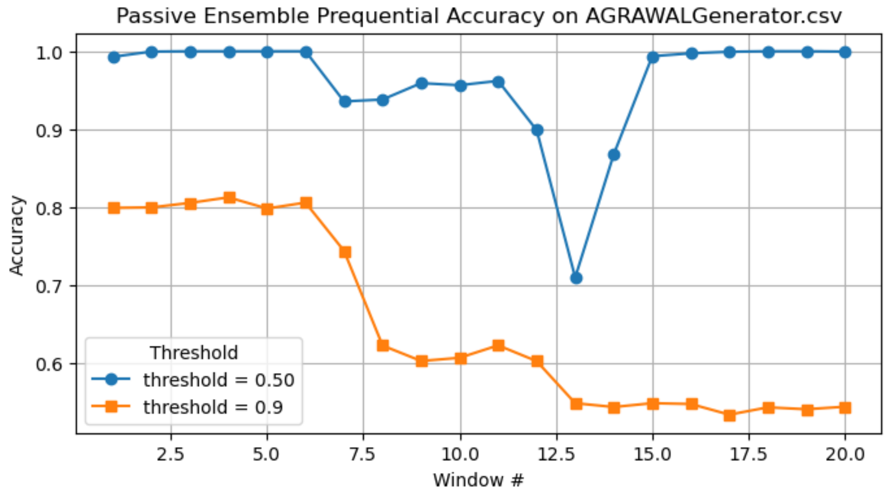

Figure 9 Demonstration of the influence of threshold in passive drift ensemble model

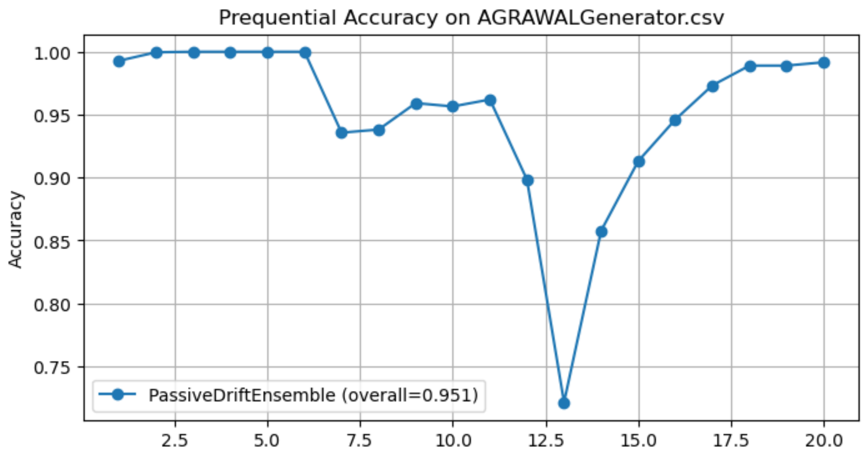

Figure 10 Prequential Accuracy as a function of window number on AGRAWALGenerator.csv

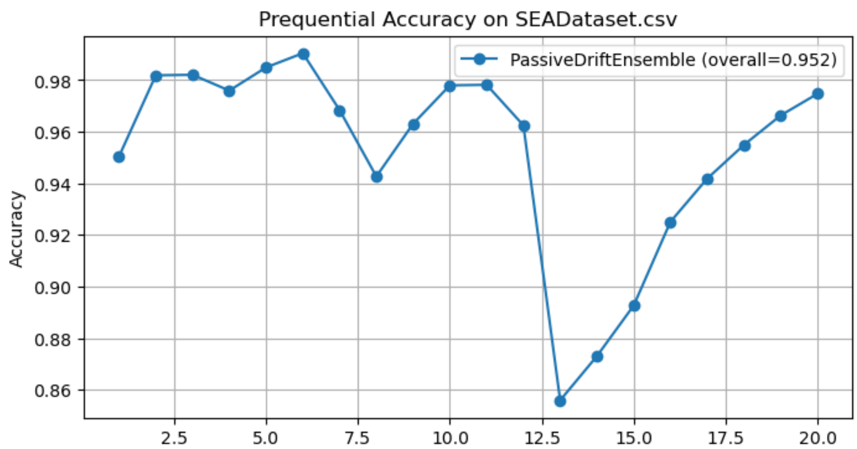

Figure 11 Prequential Accuracy as a function of window number on SEADataset.csv

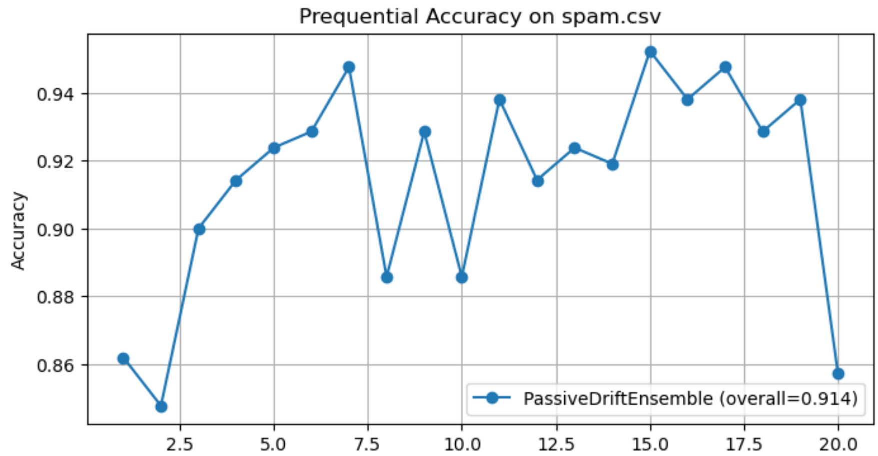

Figure 12 Prequential Accuracy as a function of window number on spam.csv

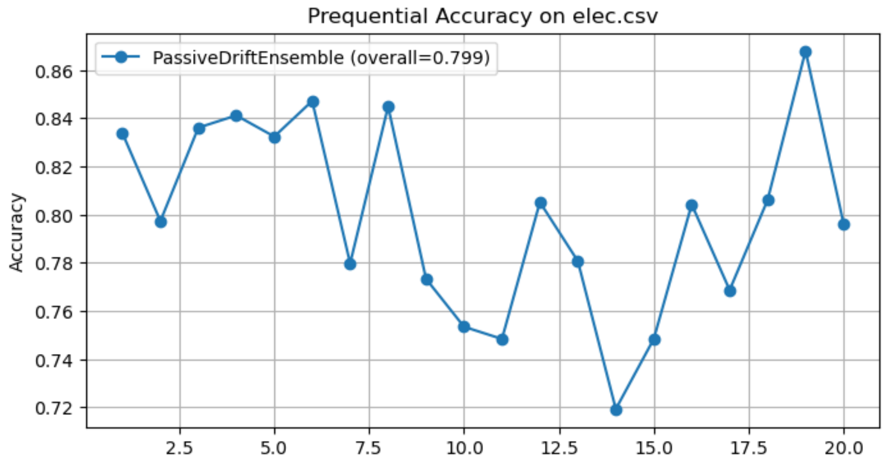

Figure 13 Prequential Accuracy as a function of window number on spam.csv

5. Results and Discussion

5.1

As mentioned in 4.1 the sudden drops in prequential accuracy plots indicate the concept drifts in the data and the swiftness of the recovery of the model determines it's adaptibility. In Table 1 all of the overall accuracies is provided for 4 different models and 4 different datasets which makes in total 16 cases.

Model/Dataset	AGRAWALGenerator.csv	SEADataset.csv	spam.csv	elec.csv
ARF	0.945	0.991	0.955	0.852
SAM-knn	0.574	0.968	0.948	0.782

Active Drift Ensemble	0.917	0.976	0.914	0.848
Passive Drift Ensemble	0.951	0.952	0.914	0.799

Table 1 Overall accuracies of all models on each dataset

Altough the ensemble models are close to the state of art models, ARF outperforms all of the models. Moreover, state of art models have a swifter adaptibility compared to custom ensemble models. There are several methods to make the ensemble models even closer to state of art models in terms of accuracy. An idea is that, hybried drift detection. Active and passive detectors can be used at the same to detect drastic concept drifts with active detectors and as well nullify the effect of false alarms with passive detectors. Another approach is that in this assignemnt on Hoeffding trees are used as base learners however additionally different base learners can be used, such as Naive Bayes, to enhance the model performance.

5.2 The window size is important size it determines the adaptability of the model. A larger window size provides less noise however it does not swiftly respond to concept drifts. On the other hand small window sizes are swift in terms of responsiveness to concept drift however is more noisy as the tradeoff. Hence a moderate window size of 150 is fixed throughout the assignment.

5.3 If the average of overall accuracy over 4 datasets is considered, active drift ensemble model outperforms the passive drift ensemble. To understand the underlying reasons several factors should be considered. Consider Figure X and Figure Y, the SEADataset.csv is known to have concept drifts around window 7 and window 12 which corresponds to 35000 and 60000 points, the passive ensemble takes more samples to get back to the previous best accuracy which means it reacts slower to concept drift when compared to active drift ensemble which explains the high accuracy of active ensemble model. Other parameters such as noise, stability, false alarms can be observed from the following table. The results can be generalized for all datasets.

Ensemble/Result	σ	Total reset of base learners	False replacement of base learners
Active	0.0202	45	%22.7
Passive	0.0178	13	%13.1

Table 2 Desired result for the comparison of active and passive drift ensemble models

As the table demonstrates the drift detection mechanism of active drift ensemble model is more volatile. Due to volatiliy the active drift mechanism replaces more learners and has a higher rate of false replacement which makes it relatively less stable and noiser compared to passive drift detection. However this trades off with swifter adaptibility of the model and the swifter adaptibility provides higher overall accuracy rate compared to passive drift ensemble model. To improve the accuracy of the passive drift ensemble model it should be more adabtable to concept drifts which can be achieved through window size and threshold tuning. One solution is to decrease the threshold or decrease the window size slightly for swifter adaptibility however the tradeoff with accuracy should be considered. If the paramter values are dropped drastically accuracy eventually drops.

5.4 This assignment is benefitial in terms of grasping data stream, concept drifts, prequential evaluation and implementation of passive and active drift ensemble models. The assignment brodened my vision in handling the concept drifts with the implementation of passive and active drift ensemble models. Furthermore how each parameter influences the adaptibility, accuracy of the model is educating. After completing the assignment, due to the new experience and knowledge I feel more confident about data streams and concept drifts.