The classifiers used varied from being the simplest (and thus inefficient) to some good performance algorithms (computationally expensive). Methods like No-change classifier and majority classifier were the most inefficient as the dataset varied for different target classes and with other attributes. The classifiers relied almost purely on chance to get better accuracies(as we can observe with their random peaks).

Methods like Hoeffding Trees, SAM-KNN and Hoeffding Adaptive Trees on the other hand performed consistently well and plateaued at a certain accuracy after the training data.

Hoeffding Adaptive trees performed slightly better in the long run compared to the vanilla hoeffding trees due to the use of ADaptive WINdowing(ADWIN). The drift detection analysis helps boost the accuracy of this classifier.

Out of all the classifiers used, leverage bagging has produced the most efficiency.

Analysing this data helps us realise that different classifiers perform differently for different datasets. For example, dataset like UV index of a city don’t drastically change over a season or day. So, a no-change classifier or majority classifier could do the job. However datasets like detecting credit card frauds require high accuracy and this a complex algorithm may be used.

**Comparison of results with reference paper**

**No-change classifier**

|  |  |  |
| --- | --- | --- |
|  | Paper(Accuracy %) | result obtained(Accuracy %) |
| Abrupt | 28.98 | 27.84 |
| Incremental | 16.04 | 16.08 |
| Gradual | 38.43 | 36.19 |

Due to the simplistic nature of the algorithm, we cannot observe much difference between the paper and the personal implementation.

**Majority Class classifier**

|  |  |  |
| --- | --- | --- |
|  | Paper(Accuracy %) | result obtained(Accuracy %) |
| Abrupt | 16.07 | 31.22 |
| Incremental | 11.51 | 13.79 |
| Gradual | 15.76 | 37.26 |

Now we can observe a few differences in the accuracy percentage. This algorithm’s accuracy depends on what windows of data are being picked. This could lead to different permutations and combinations of the dataset and thus different results.

**ARF**

|  |  |  |
| --- | --- | --- |
|  | Paper(Accuracy %) | result obtained(Accuracy %) |
| Abrupt | 74.34 | 55.45 |
| Incremental | 64.29 | 57.08 |
| Gradual | 77.92 | 70.94 |

ARF results again vary time to time due to how a tree starts it formation and how the data is split. We could in fact observe differences just by running the algorithm again and again. This might be the cause to the discrepancies we observe above. Other factors like parameter tuning play a huge role too.

**Lev.Bag.**

|  |  |  |
| --- | --- | --- |
|  | Paper(Accuracy %) | result obtained(Accuracy %) |
| Abrupt | 74.34 | 69.10 |
| Incremental | 64.29 | 44.38 |
| Gradual | 77.92 | 77.26 |

This classifier gave the highest accuracies in both the paper and during experimentation

The paper uses Naïve bayes as baseline classifier for the drift detection. This experiment uses Hoeffding trees so a direct comparison cant be made. However, it is worthy to note that Hoeffding trees do perform equally better(51.28 accuracy for incremental dataset) vs naïve-bayes(52.68 for incremental dataset)