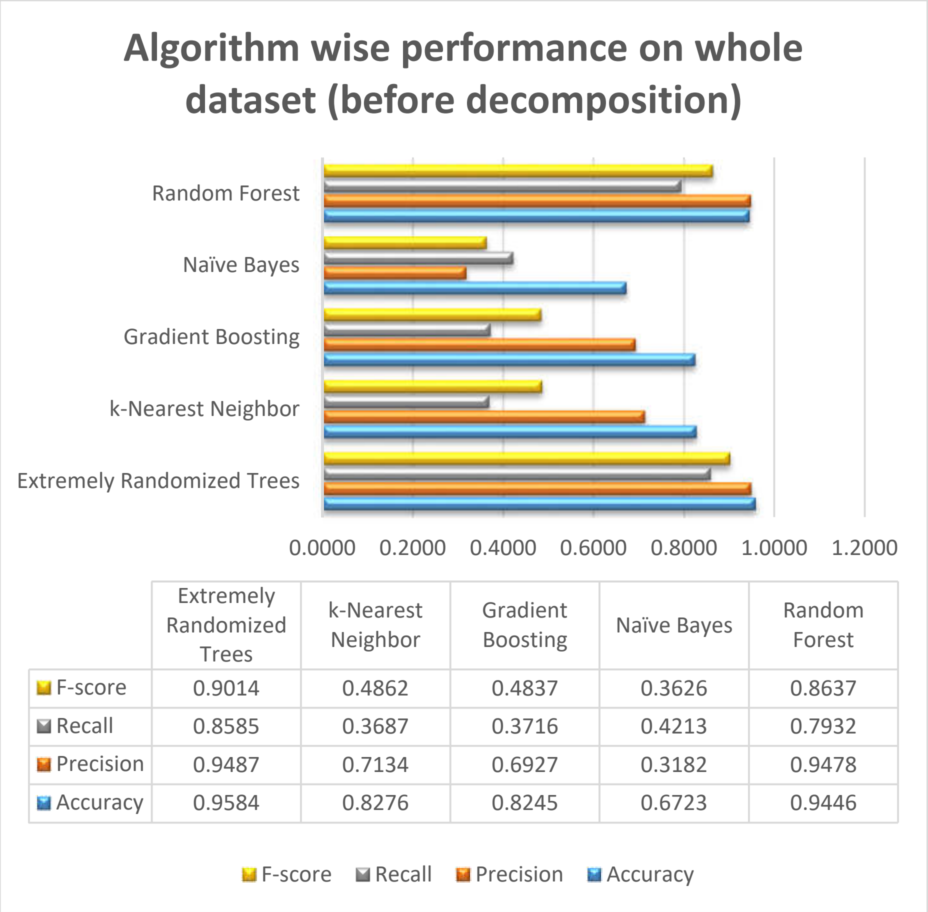
1.1 Results for Dataset II

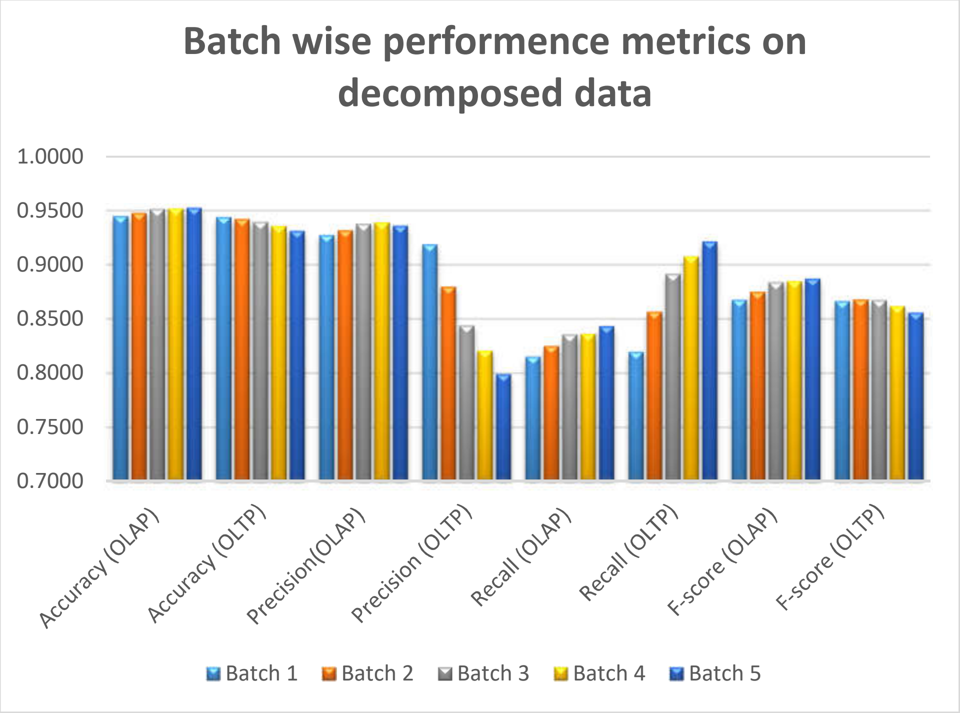
Before decomposing the dataset into OLAP and OLTP, we run different algorithms on the whole dataset. And we found that the *Extremely Randomized Trees* outperform all the algorithms in terms of Accuracy, Precision, Recall and F-score. This *Extremely Randomized Trees (ET)* algorithms, in fact, outperformed all previous state of art result[1][2] on this dataset. The performance gain is mainly due to the matter that Tree-based approach works very well for some specific type of problem where the number of features is moderate. Surprisingly we found that nobody has used this algorithm (*Extremely Randomized Trees*) before on this dataset.



	Extremely Randomized Trees	k-Nearest Neighbor	Gradient Boosting	Naïve Bayes	Random Forest
F-score	0.9014	0.4862	0.4837	0.3626	0.8637
Recall	0.8585	0.3687	0.3716	0.4213	0.7932
Precision	0.9487	0.7134	0.6927	0.3182	0.9478
Accuracy	0.9584	0.8276	0.8245	0.6723	0.9446

We divided the dataset into OLAP and OLTP data as mentioned in the *Data* section before. After that, we made 5 batches of OLAP data and 5 batches for OLTP data. We run OLAP batch 1 and OLTP batch 1 serially. And this way OLAP 2 and OLTP 2 serially next. By this way at the end of batch 5, result from OLTP- batch 5 is the final result from the decomposed dataset.

	Accuracy (OLAP)	Accuracy (OLTP)	Precision(OLAP)	Precision (OLTP)	Recall (OLAP)	Recall (OLTP)	F-score (OLAP)	F-score (OLTP)	Execution Time (OLAP)	Execution Time (OLAP)
Batch 1	0.9450	0.9441	0.9273	0.9189	0.8153	0.8196	0.8677	0.8664	13.0363	166.1613
Batch 2	0.9479	0.9424	0.9318	0.8797	0.8250	0.8565	0.8752	0.8680	13.21439	190.834
Batch 3	0.9514	0.9396	0.9379	0.8441	0.8356	0.8917	0.8838	0.8672	13.48821	193.0443
Batch 4	0.9518	0.9356	0.9391	0.8205	0.8363	0.9076	0.8847	0.8618	13.02166	176.3571
Batch 5	0.9526	0.9314	0.9361	0.7990	0.8433	0.9215	0.8873	0.8559	16.66352	187.3037

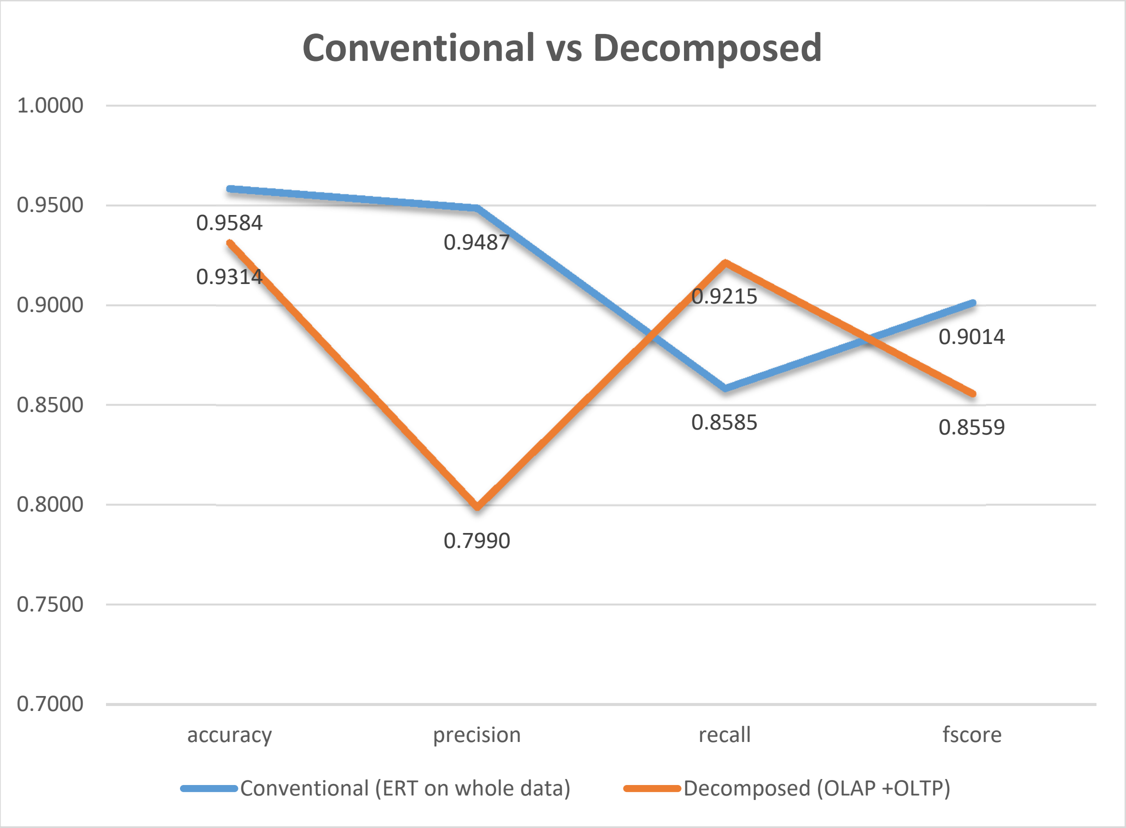


From above figure, we can see that **recall** for both OLAP and OLTP is increasing with the increase of the batch number. Which imply that the percentage of target (default) detection rate has an increasing order with the batch number.

	TP	TN	FP	FN	accuracy	precision	recall	F-score
Conventional (ET on whole data)	5697	23056	308	939	0.9584	0.9487	0.8585	0.9014
Decomposed (OLAP +OLTP)	6115	21826	1538	521	0.9314	0.7990	0.9215	0.8559

The above table and the figure below is the comparison of performance using conventional approach (applying the best classifier, Extremely Randomized Trees on the whole dataset) and our proposed framework (Decomposing the data into OLAP and OLTP). We want to mention that both of our approaches outperformed the state of the art result [1][2] on this dataset. So far we have seen a maximum accuracy of 81.96% and maximum Recall of 65.54% in all previous research works on this dataset. Whereas for both of our approach the accuracy is above 93.13% with a maximum of 95.84%. We have gained a better recall percentage too. The recall is also called True Positive Rate(TPR) for binary classification. In fraud or risk, detection Recall is very important for overall benefit. Though for maximizing recall introduce an increase of False Positive Rate, which is normal for risk analytics.

Another mentionable contribution of this research using the decomposition approach is the early detection. If we notice the recall of all batch (1 to 5), we can see that in the first batch 81.96% of risky accounts were detected. And from the data of 5 months later, which means batch 5 we could detect 92.15% of risky accounts. So this early detection could help organizations to avoid a great amount of loss.


Conventional vs Decomposed
accuracy
precision
recall
fscore
Conventional (ERT on whole data)
Decomposed (OLAP +OLTP)