III)Experiment

The experimental environment is processed centrally on a computer configured with Intel(R), Core(TM) i7-6820HQ CPU @ 2.70GHz (8 CPUs), 2.7GHz and Windows 10 operating system. In this paper, there are 5 multi thread experiments on 5 dataset. The Accuracy or amount rules will be skipped because this paper focus on reducing running time by using multi thread on ACAC.

1. Default of credit card clients Dataset

The first experiment is implemented on Default of credit card clients [1]. It has 23 columns and 30000 instances in each columns. In that dataset the last attribute is a target attribute named “default payment next month” to classify Default of credit card clients as 0 or 1 (Yes = 1, No = 0). The reason why this paper use multi thread ACAC for that first dataset because the number of that experiment enough large to optimize execute time.

Table I. The result of multi threads on ACAC in Default of credit card clients dataet

Number of threads	Times(ms)
1	564
2	463
3	419
4	476
5	512

Table I describes the running time of ACAC on each number of threads. In this dataset, the best running time at number of threads = 3.

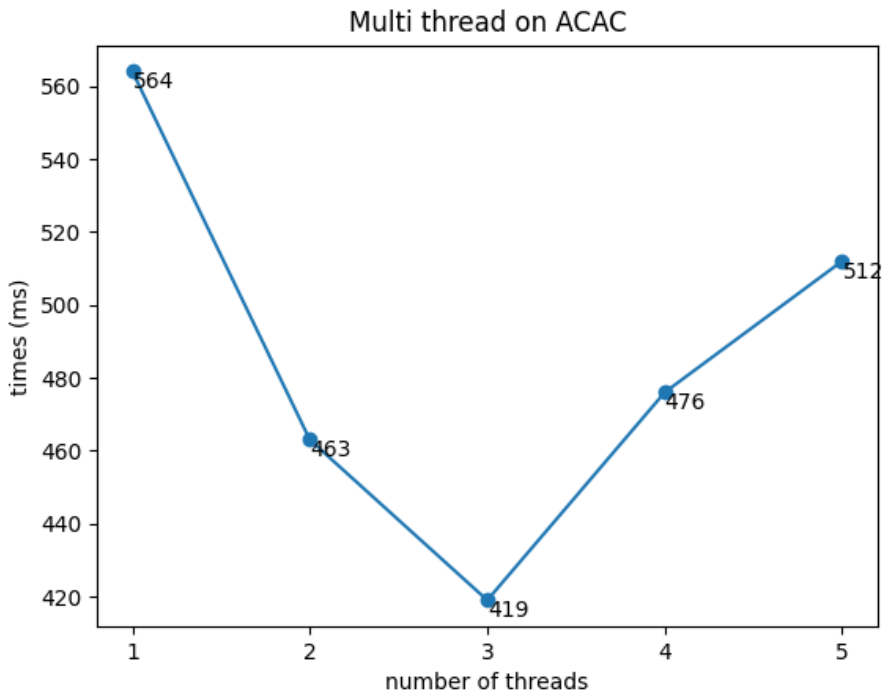

Figure 1: Times run of ACAC on each number of threads in Default of credit card clients

In figure 1, the horizontal axis depicts the number of threads used in this experiment, the verical axis depicts the time run of ACAC. The graph reach highest running time(worst) at using 1 thread on ACAC in that dataset. on the other hand, the expectation result of this paper will be reached at using 3 thread on ACAC.

2. Mushroom Dataset

The second experiment is implemented on Mushroom Dataset[2] the classic data for every classification algorithms. In Addition, Mushroom is also used by original ACAC algorithm, it has 8125 instances and 23 columns. In the data, the first attribute is a target attribute named 'class' to classify mushrooms as poisonous or non-poisonous (edible=e, poisonous=p). The another reason why this dataset is chosed because of its amount instances. As previous presentation, the throwback when Using multi thread on ACAC is the number of instances. The running time of ACAC will be optimal if there are huge instance, because multi thread will spend time on Creating thread in evaluating every rule.

Table II. The result of multi threads on ACAC In Mushroom Dataset

Number of threads	Times(ms)
1	1187
2	1509
3	1792
4	2061
5	2315

Table II describes the running time of ACAC on each number of threads. In this dataset, the best running time at number of threads = 1 (non using multi threads).

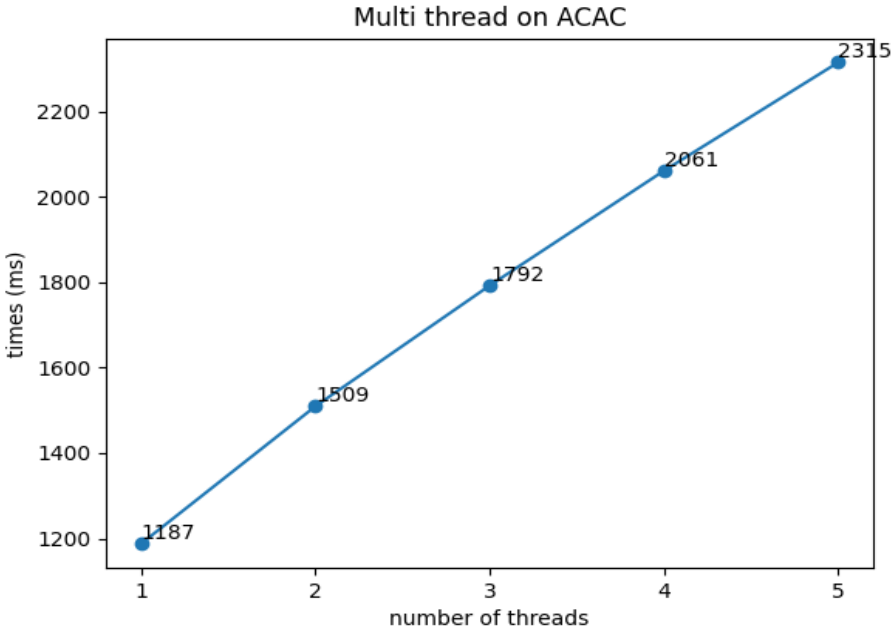

Figure 2: Times run of ACAC on each number of threads in Mushroom

In figure 2, the horizontal axis depicts the number of threads used in this experiment, the verical axis depicts the time run of ACAC. The graph reach highest running time(worst) at using 5 thread on ACAC in that dataset. on the other hand, the expectation result of this paper will be reached at using 1 thread on ACAC.

The main reason why this dataset get unexpectation result because Mushroom dataset does not contain enough instances for using multi thread on ACAC

2. mail spam Dataset

To clarify the throwback of multi thread on ACAC, this paper continues the experiment on mail spam[3]. This dataset has 58 columns and 6025 instances, the first attribute is a target attribute named 'spam' to classify a mail as spam or non-spam (spam = 1, non-spam = 0). But 58 columns with a huge value of each column, so this paper used a new A meta-heuristic approach for enhancing performance of associative classification[4] for preprocesssing to reduce worse columns, which don't dedicate for classification.

Table III. The result of multi threads on ACAC In Mail spam Dataset

Number of threads	Times(ms)
1	1187
2	1509
3	1792
4	2061
5	2315

Table III describes the running time of ACAC on each number of threads. In this dataset, the best running time at number of threads = 1 (non using multi threads).

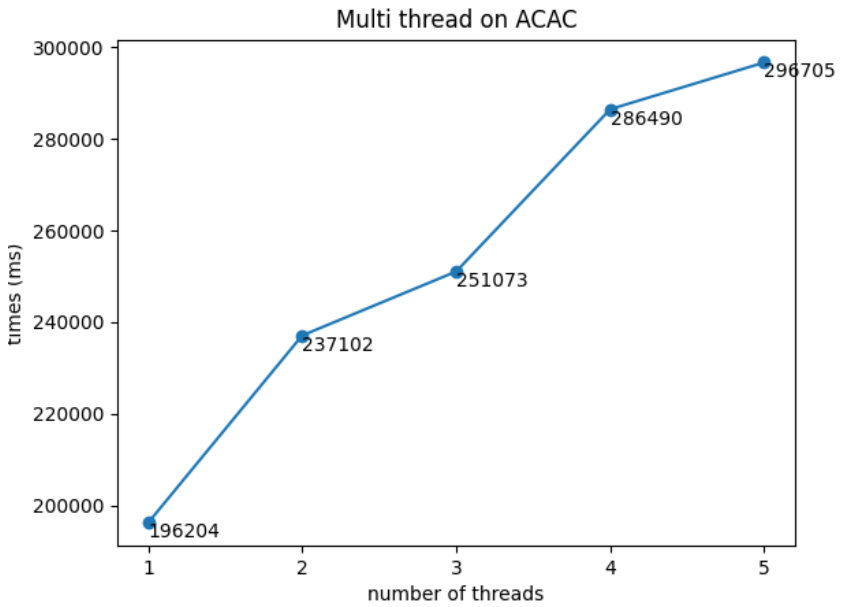

Figure 3: Times run of ACAC on each number of threads in mail spam

In figure 3, the horizontal axis depicts the number of threads used in this experiment, the verical axis depicts the time run of ACAC. The graph reach highest running time(worst) at using 1 thread on ACAC in that dataset. on the other hand, the expectation result of this paper will be reached at using 1 thread on ACAC.

IV)Conlusion

The ACAC algorithm is only effective on data sets with few instances like Mushroom[2], Mail Spam[3]. For data sets with many instances like Default of credit card clients Dataset [1] the algorithm exhibits limitations. This makes the ACAC algorithm in particular and CAR algorithms in general impractical when classifying data, while CAR algorithms are a type of classification algorithm that is easy to explain (explainable) the reason for classification.

This approach cannot be used for the purpose of reducing bad rules or gaining accuracy, it focuses on reducing the runtime evaluating every rule in the Algorithm, so for sets For data with large attributes, this method is shown to be ineffective.

For algorithms with a small number of data set reads, it is not suitable to use this approach because multithreading focuses on reducing the time to evaluate a rule, the larger the number of data reads, the more The faster the time stream for evaluating a rule will be. In other cases, algorithms with a small number of data reads will not be suitable for this approach, and a low number of multithreads will not improve the processing time. Some algorithms that should not use this method include ID3, Decision Tree, CMAR, families of algorithms for building decision trees.

V)References

[1] https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients
[2] https://www.kaggle.com/datasets/uciml/mushroom-classification
[3] https://www.kaggle.com/datasets/yasserh/spamemailsdataset
[4] H.Q.Huy, T.A.Tuan, D.N.Tai, “A meta-heuristic approach for enhancing performance of associative classification”, 2024-01-22.