## DP-GBDT Algorithm Adaptation

The idea, “Theos hypothesis”, is as follows:

When training normal (non-DP) GBDTs, the predictions get closer to their true values with every new tree that is added to the ensemble. However, due to the randomness introduced by DP (when nodes select a split value using the exponential mechanism), this is not always the case for DP-GBDT. So some of the trees might actually make the final prediction worse. Therefore Theo decided to set aside some samples/instances that are used each time after a tree is created, to judge whether it’s a good/useful tree.

Algorithm:

* The data is first divided into train and test sets, A and B.
* The Train() function is then called on set A.
* Inside the Train() function there is a second split into A1 and a smaller A2. Then A2 is set aside to later determine if a tree (grown using his share of A1) is a “good/useful” tree.

This seems to massively improve accuracy. However, since A2 is reused it’s not DP in the current setting of Theos code. So some privacy/proof adaptations would have to be explored.

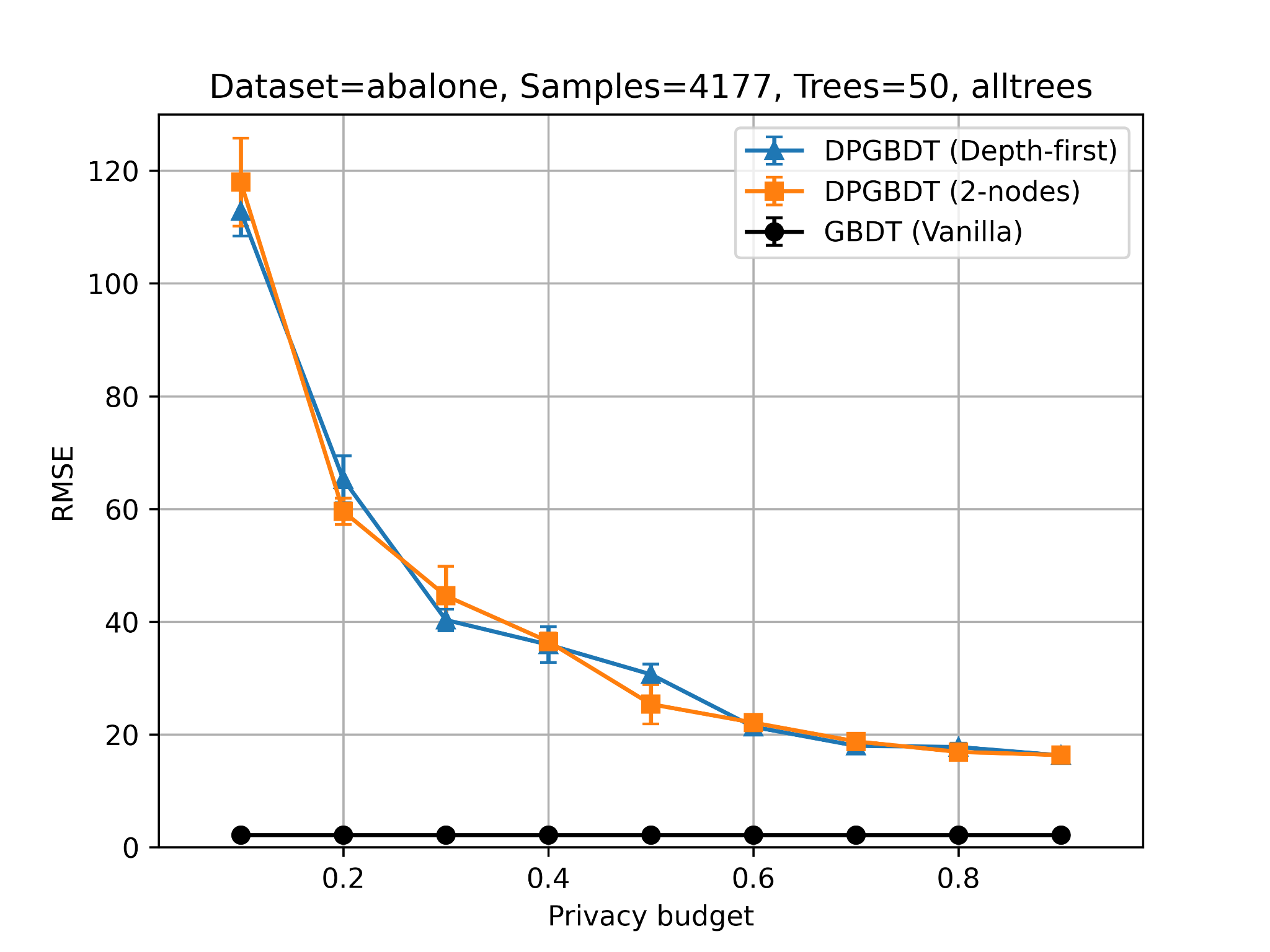
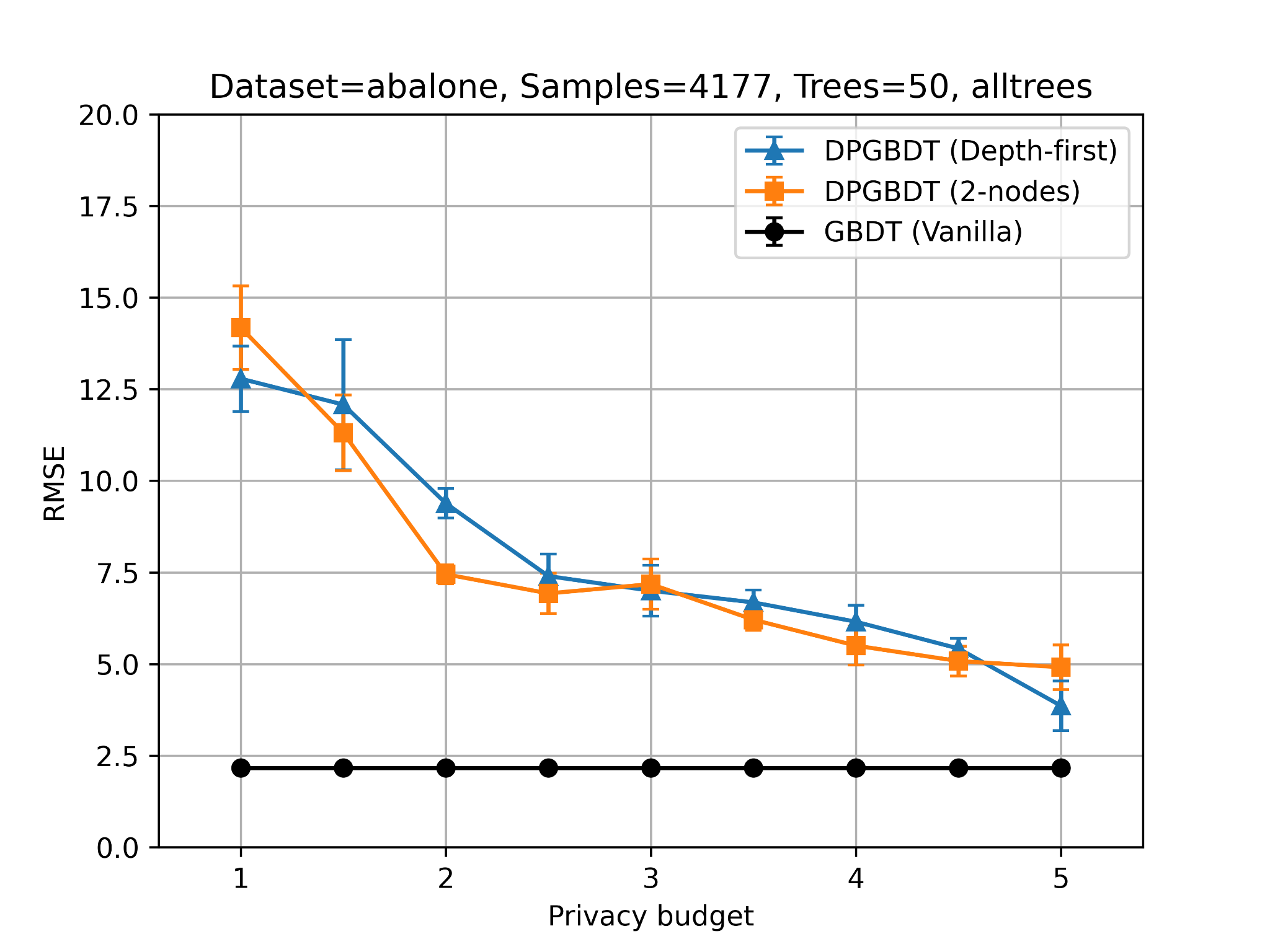
Another idea, that would probably work given enough data, is to split A2 again into num\_trees shares. So each tree could be accepted/rejected using a separate share. This could be easier to prove.

### Graphs (abalone.data):

“Textbook” DP-GBDT, no 2nd split

We don’t divide into A1 and A2, and use all (in this case 50) trees for prediction. 1 ensemble with 50 trees. 4177 samples, 5-fold cross validation -> 66 training samples per tree.

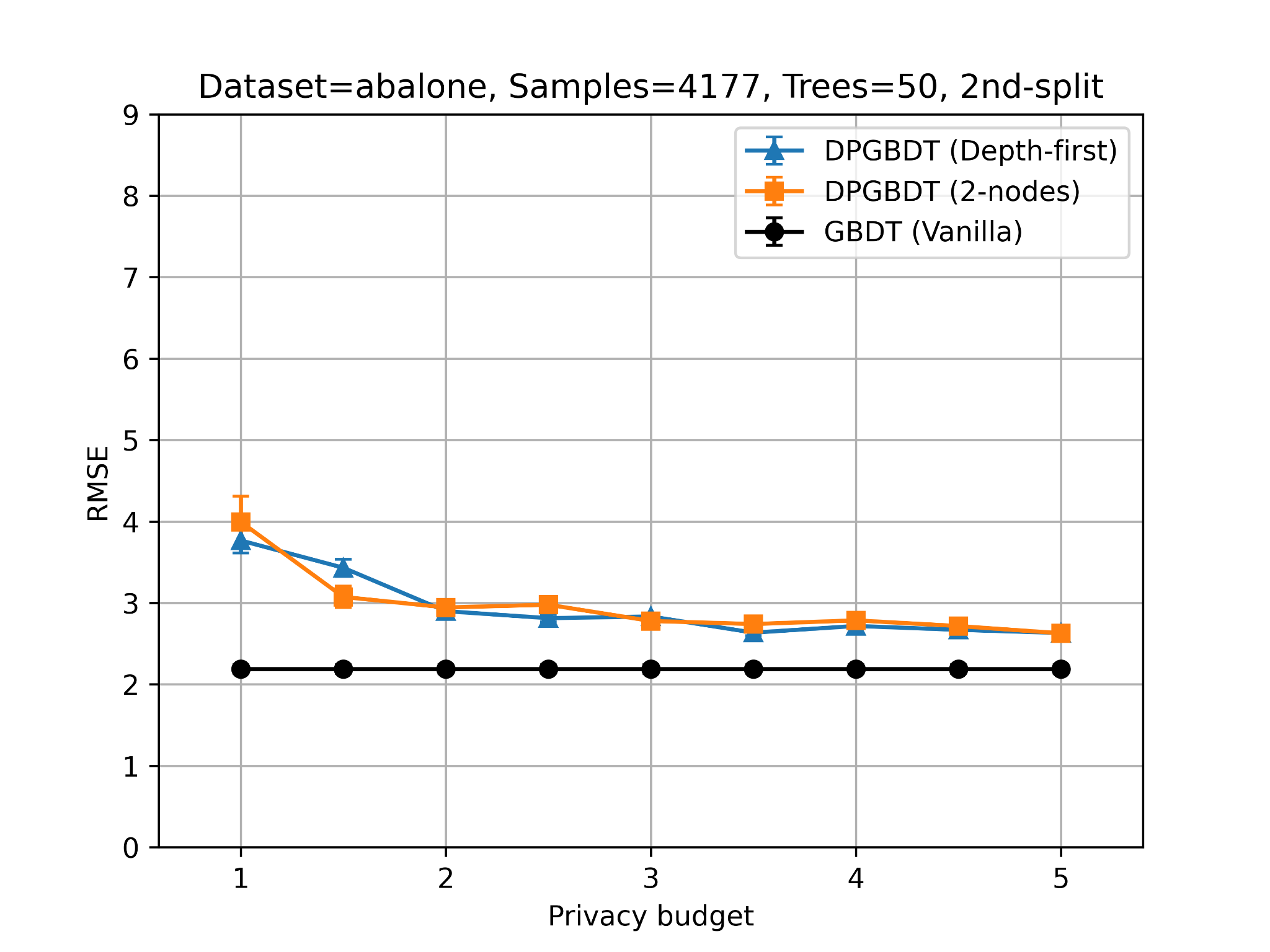
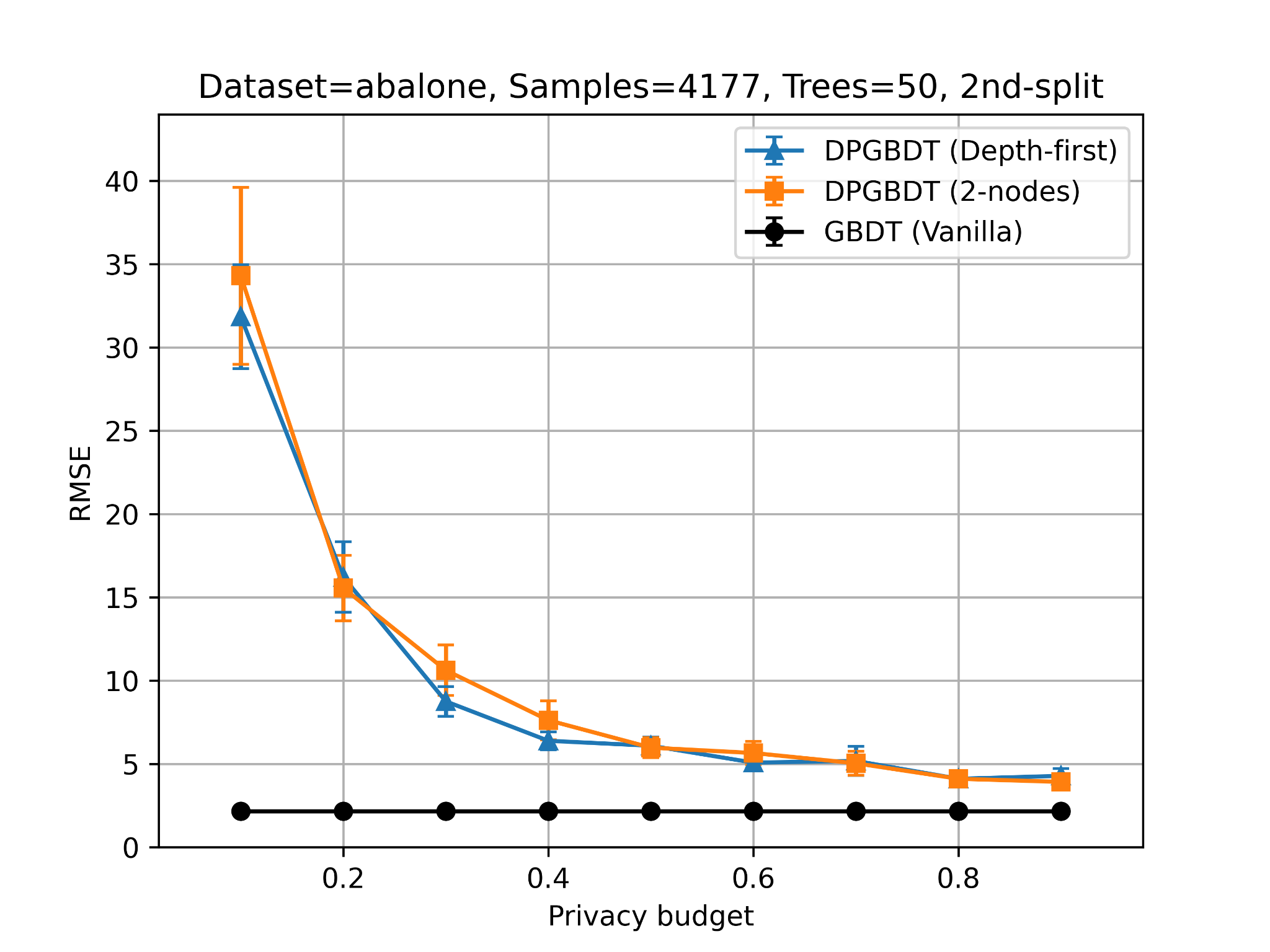
The plot is divided into 2 graphs because of the large RMSE variations.

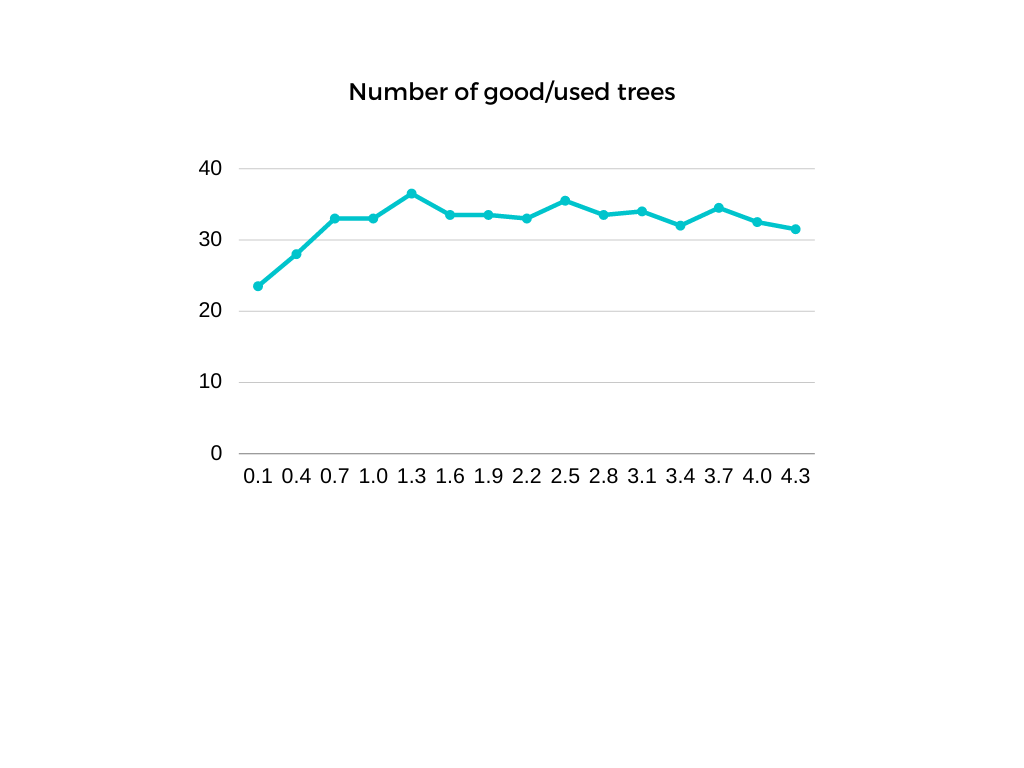
“Theos hypothesis”, 2nd split DP-GBDT

Use A1 to grow the trees and repeatedly use A2 for accepting/rejecting.

1 ensemble with 50 trees. 4177 samples, 5-fold cross validation -> 46 training samples per tree. The plot is divided into 2 graphs because of the large RMSE variations.



Depending on the privacy budget (and randomness), around 20-40 trees out of 50 are used for prediction:



I created a separate directory with the different codes to create the measurements above:

<https://gitlab.inf.ethz.ch/kkari/enclave-hardening-ML/-/tree/master/code/theos_hypothesis>