The dataset we are using is MNIST with 60000 training samples and 10000 testing samples. The task is to classify 28*28-pixel images of hand-written numbers into its corresponding digit.

The experiments are done on a fully-connected neural network with 2 hidden layers in size 800 and 500 respectively, with an output size 10, one for each class. We simulate the distributed procedure with $N$ threads, each representing a worker, among them $s$ are faulty agents that would calculate the correct gradient, reverse it and time 100. For each epoch, there would be $s$ workers randomly selected to be faulty.

The 60000 training samples are owned by all $N$ workers from the beginning. Each step every worker will calculate the average gradient of a $b$-size batch of samples from their own 60000 samples, the drawing order is randomized and different for each worker, such that they are not using the same batch at the same step.

In the following experiments, set $N = 40$, $s = 15$, $b = 128$. Each epoch contains $60000/128 = 469$ batches, of which the last batch only contain 96 samples.

The filters that are used are averaging (normal) and krum.

The loss function is cross entropy. The results are evaluated using precision@$k$, where if among the first $k$ results there is the correct answer, the prediction is counted as true positive. Here $k = 1, 5$.

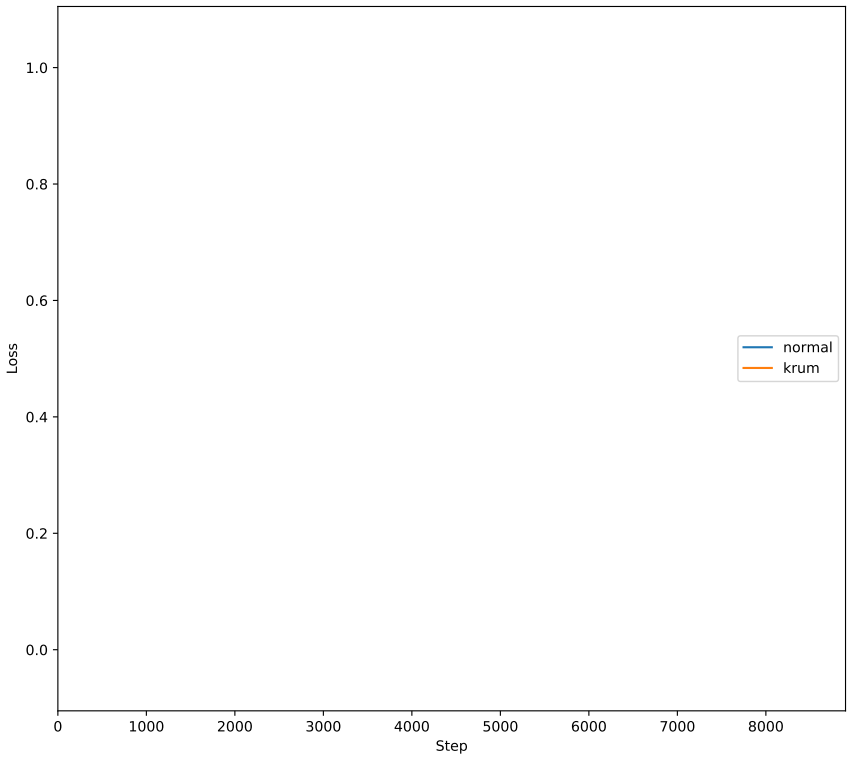


Figure 1: Loss

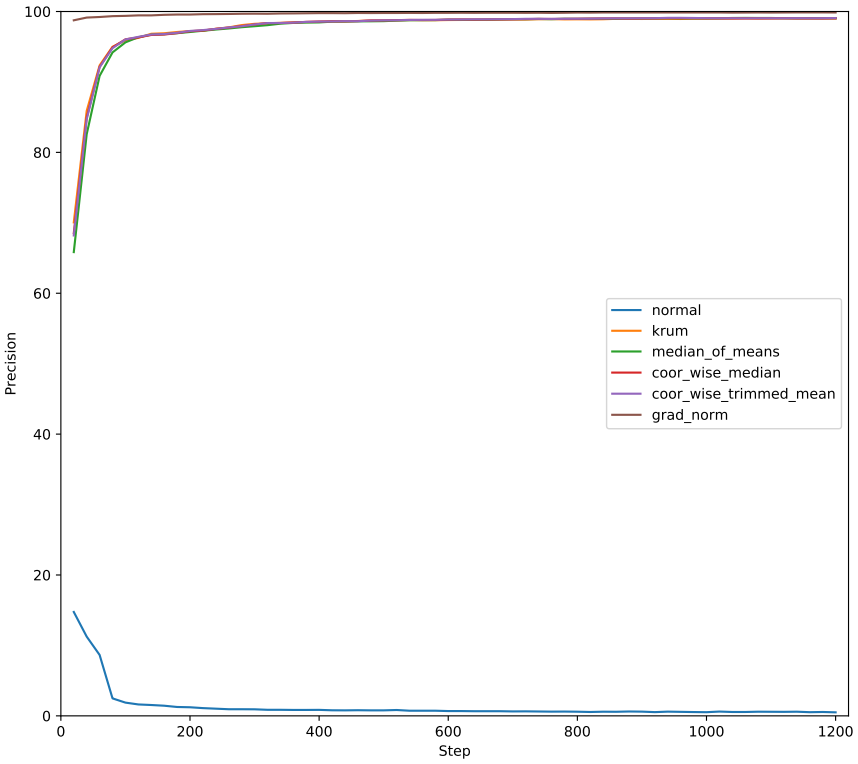


Figure 2: Precision@5