Measurements: Load [%], Power [W], temperature [C] and so on for CPU and GPU. Query length, type, connection state, time, model,etc.

To interpret the results better let us introduce **new metrics**.
=================
Good **distribution**. (table)

Spread because of **outliers** or as it is.

Hard topics
=================

Different models spend different energy

Local computations are much much higher (**x10**). Supports **the paper**

=================

The results are “**groped**” for all models. So, **length** of the query is much more important than **internet** connection condition.

We can say that energy consumption depends not so much on the **number of parameters**, but on the type of **model**, its operation, etc. The trend can be- but this is just not the most important.

=====================
We can see what we already saw: **local computations** results are **higher**

and the **less** is a **request**, **less energy** is spended.
====================

This “hill” may be due to **connection quality adaptation algorithms**: transcoding (often changing encoding), delay and error handling (FEC/PLC), analysis of lost data, and sending retransmission requests. These jumps in transmission level increase/decrease require significant computational effort. Dynamic Quality Adjustment, Congestion Control algorithms, Adaptive Bitrate Streaming, etc.

Slightly **faster** than with a bad connection, but with much **more CPU Power**.

==========================

The same **distance**

Powerfully (with big **consumtion**) and **quick**; or economically (with small consumtion) and slow. More consumtion per hour, but less hours; or less consumtion per hour, but more hours.

===========================

For server side results for **medium connection** is also **higher** than for normal and bad. We think this is because it has to **resend** packages

Again: worse the **condition**, easier the **load**, less the **energy consumption**.

For **local computations**, the costs are even higher.

Longer time.

Writing locally, **more time in active state**.

The result is again **less time**, but **higher load**, when using Wi-Fi, and more **time**, but **less load**, when writing locally.
But now the first approach is **slightly** better.

*Also perhaps **GPU is loaded** more for local one.*

It can be better consider **more measurements** and **components** for this (SSD, WiFi chip, etc).

—---------------

*As we understood it is because of the specifics of writing data locally and sending it over the network: when writing locally, the **CPU operates less**, but the writing itself **takes longer**, as it requires searching for a location and writing it to disk.*

*Furthermore, as we understand, writing occurs in **small portions** (word by word), which results in many small steps of writing. Meanwhile, when sending over the network, the CPU is more heavily loaded (due to encryption, copying buffers, etc.), but the **sending occurs immediately and faster**. Furthermore, the **Wi-Fi chip** takes over some of the work. After sending, the Wi-Fi module quickly goes into a low-power state, and the **CPU returns to idle**. Components do not go into sleep mode while writing to disk, while they can when sending over the network.*

================================

Short answers are more **efficient**.

Medium is the worst.

================================
Conclusions
================================

We considered systems, which "**shut down**" at the end of response, as in paper, as we got it. More realistic can be an approach that will also consider **time of working** after a request. But we expect the normal connection to be even worse than the bad one. With less time period we had more or less same results for normal and bad connections, so adding some more energy for spended time for **normal connection** can make its result **even worse**.