Using ChatGPT to analyze large CAN data sets

In a recent tutorial by CSS Electronics, it has been shown how one may leverage ChatGPT (GPT4) with 'Code Interpreter' to analyze large amounts of CAN data.

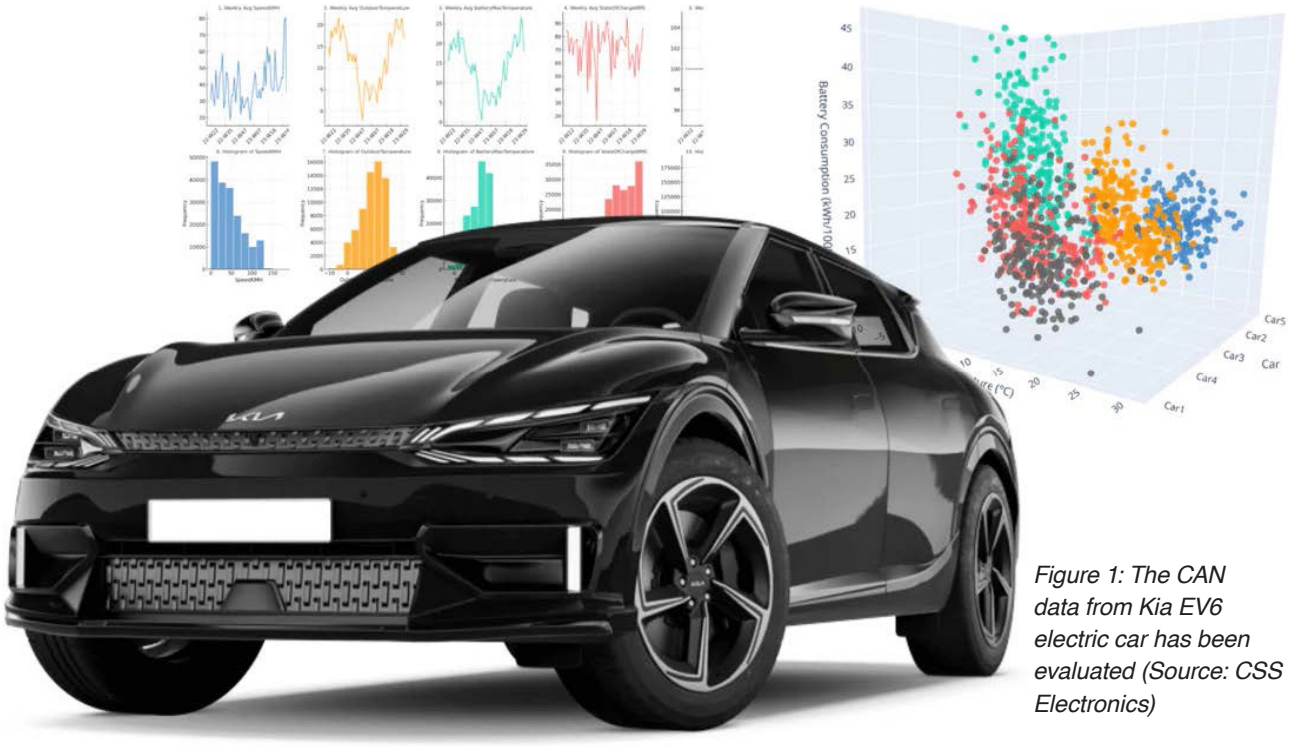


Figure 1: The CAN data from Kia EV6 electric car has been evaluated (Source: CSS Electronics)

CSS Electronics uses a dataset from a Kia EV6 electric car, which was recorded using a CANedge3 CAN data logger with an SD card and 3G/4G connectivity. The data consists of Unified Diagnostic Services (UDS) data and GPS/IMU data from the internal sensor of the CANedge3. More than one year of raw CAN data was decoded using the Python API (application programming interface) and DBC (data base CAN) files, resulting in a 100-MiB CSV-file with more than 80 signals, ready for analysis.

By using a ChatGPT Plus account (20$/month), the team enables the 'Code Interpreter'. This allows for uploading the data files (e.g. CSV) and asking the chat bot to analyze the data by creating and executing the Python code. ChatGPT directly outputs the code as well as the results e.g. as images, HTML files, etc.

The full article available from CSS Electronics explains how to prepare the CAN data and provides 12 example show cases. For each example, the company shows what prompt they use and the resulting output from the ChatGPT. Below, the company recaps four of the 12 examples.

Example #1: Creating 10 insightful data visualizations

The first example shows how ChatGPT can be asked to visualize the data in different ways. The team initially asks the chat bot to come up with 10 insightful data visualizations on its own - but this type of high-level task results in rather random visuals. However, by instead asking it to produce a specific plot with specified signals and charts, ChatGPT very effectively does this with a minimal follow-up.

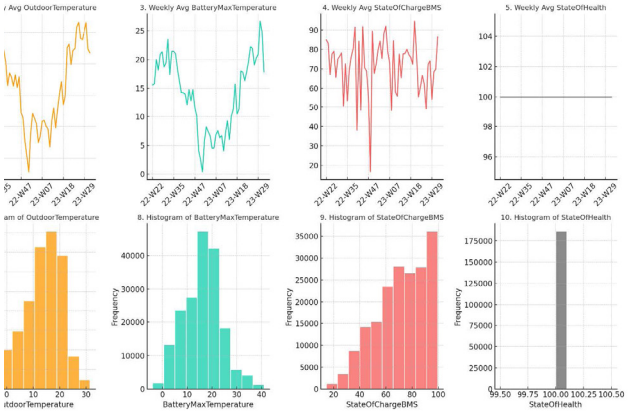
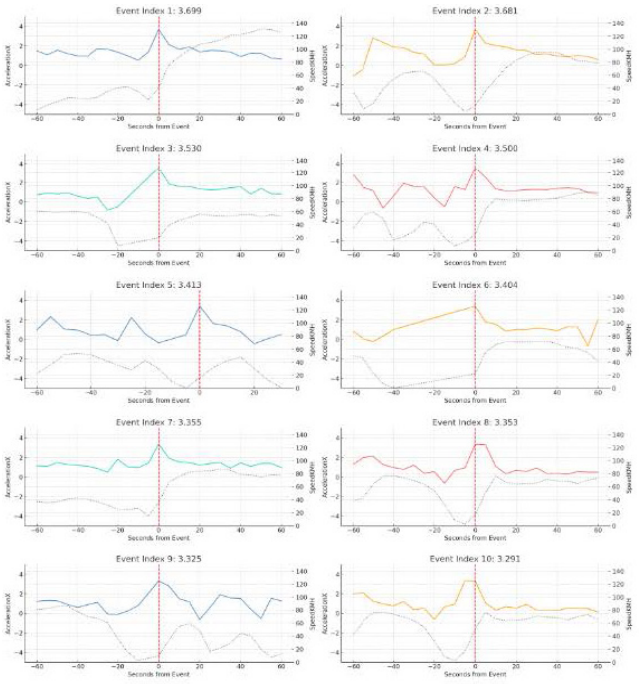
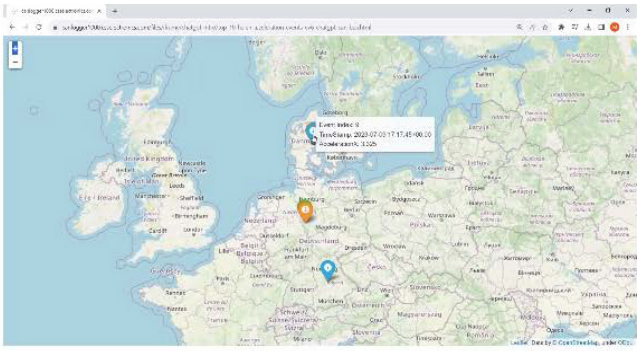


Figure 2: Plots produced by ChatGPT for specified signals and charts (Source: CSS Electronics)

Example #7: Visualizing top 10 acceleration events

When analyzing large amounts of data, it is often relevant to look for specific events. To illustrate this, the article shows how to identify the top 10 acceleration events of the electric ▷





Event Index	TimeStamp	AccelerationX
1	2023-07-18 14:35:20+00:00	3.699
2	2023-07-17 15:02:10+00:00	3.681
3	2023-07-06 15:12:40+00:00	3.5302
4	2023-07-09 17:18:50+00:00	3.5
5	2023-07-06 06:52:20+00:00	3.4125
6	2023-07-09 17:22:50+00:00	3.4036
7	2023-07-17 14:45:20+00:00	3.3546
8	2023-06-30 11:14:00+00:00	3.3534
9	2023-07-09 17:17:45+00:00	3.325
10	2023-06-30 11:14:05+00:00	3.2908

Figure 3: Plots produced by ChatGPT for top 10 acceleration events (Source: CSS Electronics)

vehicle, under the restriction that there must be a certain number of observations before/after the event - and that the vehicle speed must be non-zero.

ChatGPT successfully produces a plot of each acceleration event (see Figure 3). In addition, it manages to visualize each event in a geographical plot as requested by the user's prompt. Notably, all of these outputs were produced in about 15 minutes - with zero knowledge required from the user.

Example #11: Who's the better driver - Martin or Josefine?

In perhaps the most complex request, ChatGPT was asked to perform a multi-step analysis to benchmark the driving behavior of two drivers of the EV6, Martin and Josefine.

To segment the data by the driver, ChatGPT was initially asked to group the data by trips and subsequently to map each trip to a driver based on whether the car starts/ends in ▷

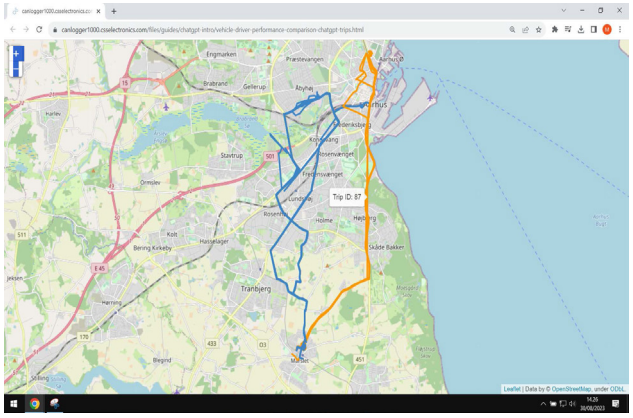


Figure 4: Visualized vehicle trips mapped to either driver (Source: CSS Electronics)

a specific geofence (thus determining if the trip was a work commute by either driver). To review the result, ChatGPT was asked to visualize the trips.

With the vehicle trips grouped, ChatGPT was asked to calculate the vehicle's power consumption and plot this in a histogram for each driver. The resulting plot shows how one driver, Josefine, manages to perform a lot better in terms of regenerative braking.

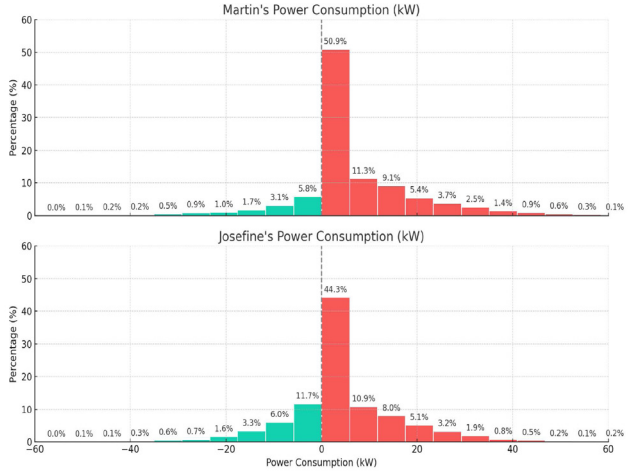


Figure 5: Visualized power consumption from Martin and Josefine (Source: CSS Electronics)

To further solidify this conclusion, ChatGPT was asked to break down power consumption into the accumulated regenerated/consumed/net amounts and to also compare the battery consumption in kWh/100km. The result clearly displays that due to regenerative braking, Josfine is driving 20-% more efficiently to work than Martin.

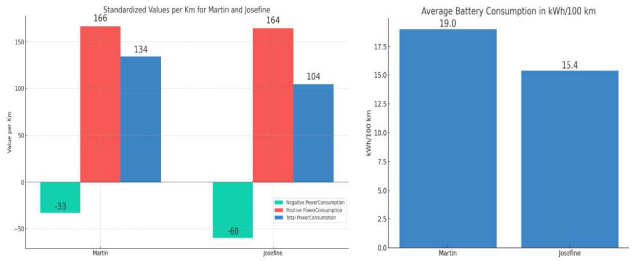
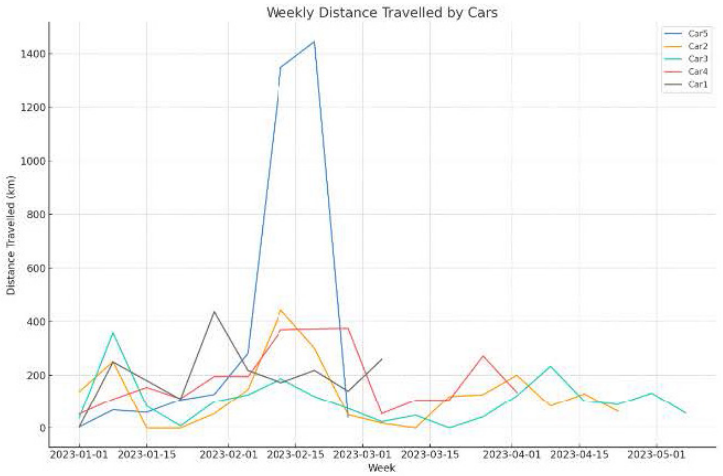
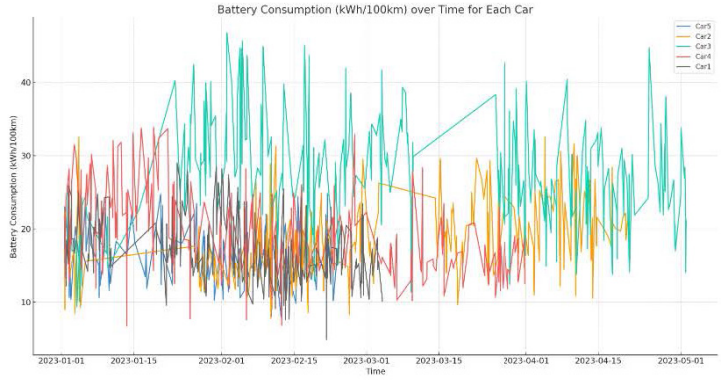
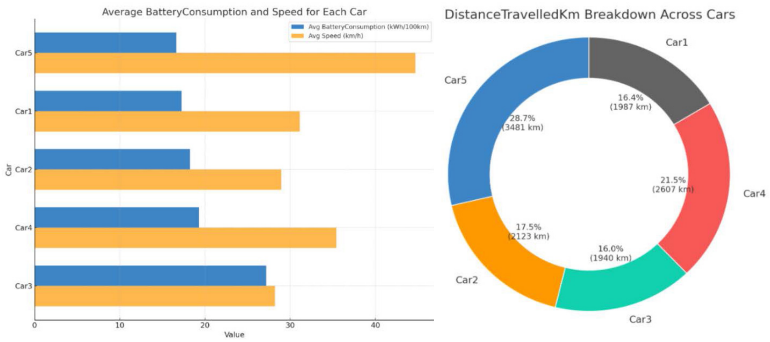


Figure 6: Compared power consumption from Martin and Josefine (Source: CSS Electronics)







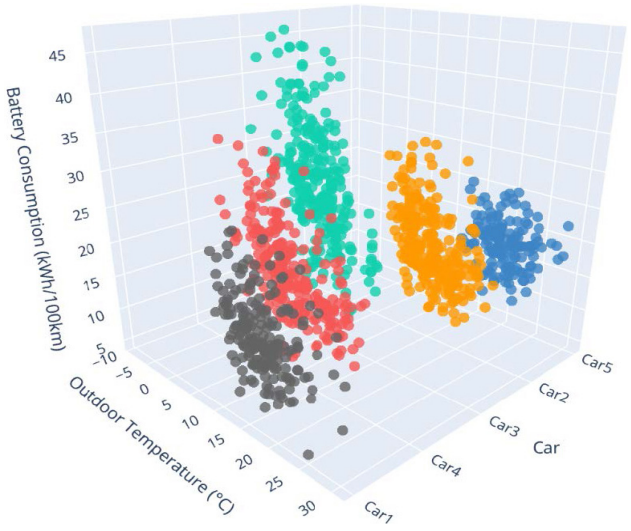


Figure 7: Data analyzed by ChatGPT across multiple vehicles (Source: CSS Electronics)

Example #12: Analyzing data from multiple vehicles

ChatGPT can also be used to analyze data across multiple vehicles and CSV files (see Figure 7). CSS Electronics illustrates this by simulating data from 5 cars and showcasing how this can be visualized in both regular charts and more advanced 3D scatter plots.

Conclusions and verdict

CSS Electronics shows that practically any analysis can be performed using ChatGPT. Using the tool does not require any coding knowledge - and it enables time savings of more than 90 %.

There are, however, also limitations. Many companies will not be allowed to upload their CAN data to ChatGPT due to data sensitivity concerns, though this may be resolved in the near future.

In addition, caution must be taken when asking ChatGPT to make conclusions based on the data. While ChatGPT generally gets the code, numbers and visuals right in almost all cases, it is still prone to hallucination when it comes to interpreting its own results. Nevertheless, CSS Electronics is convinced that ChatGPT will be a game changer for engineers and analysts working with CAN data.

Learn more in the full article or by contacting CSS Electronics.

Author

Martin Falch
CSS Electronics
contact@csselectronics.com
www.csselectronics.com