Case: Alliander

Title: Data Driven Approach towards More Efficient Newton-Raphson Power Flow Calculation for Distribution Grids

Company: Alliander

Alliander is a Dutch energy network company responsible for the distribution of electricity and gas. Operating primarily in the Netherlands, Alliander ensures the reliable and efficient delivery of energy to millions of households, businesses, and institutions. The company focuses on maintaining and modernizing its energy grids to accommodate the growing demand for sustainable and renewable energy sources. Alliander reported a turnover of €2,779 million for the full year 2023. With a commitment to innovation, Alliander plays a crucial role in the energy transition, working towards a more sustainable and resilient energy system for the future.

Use case:

With the rapid ongoing energy transition, our power grid is operating at its limit. Better utilization and allocation of resources is of paramount importance. Modelling the power grid and simulating the flow of energy are crucial for optimizing current grid usage and ensuring the grid is future-proof during expansion. Numerical computation plays a fundamental role in distribution power grid calculations. Among others, the Newton-Raphson (NR) method is the backbone of such numerical calculations.

Being an iterative method, NR is not without its limitations. It is well studied that the performance of NR depends heavily on the starting value. A good initial value could bootstrap the process and few iterations are needed for convergence, while suboptimal initial values might lead to slow convergence or even divergence.

In this workshop proposal, we aim to investigate the feasibility of a data driven approach towards NR method initial value. This fits into the problem solving pipeline where current methods include empirical user input and pilot runs of simplified methods.

We would like to invite researchers with computer science background in numerical computation, machine learning and deep learning to join us in this workshop. In particular, if you are familiar with state of the art AI frameworks (such as pytorch), or mathematical theories, or you are simply a computer scientist interested in the energy transition, please feel free to join us.

Input for workshop particpants:

We aim to provide realistic data gathered from our power grid for the workshop. Participants will be brought up to speed with our solutions and will be presented with challenging cases for discussion. By the end of the workshop, we aim to reach a tentative conclusion on the feasibility of a data-driven approach for the problem at hand. Additionally, we hope to evaluate the approach in terms of efficiency and scalability. The outcomes of this case study will be contributed to the Power Grid Model project, a widely-used open-source initiative under the Linux Foundation Energy, which is adopted by several distribution system operators (DSOs).

Cases: ING (software refractoring) and ING (legacy software)

Title case ING (software refractoring): How could AI facilitate the engineering work on software refactoring?
Title case ING (legacy software): How could AI help on understanding legacy software?

Company: ING
ING Groep NL (ING) is a global financial institution with a strong European base offering retail- and wholesale banking services. Our customers are at the heart of what we do. The future of banking will not be banking as we know it. Technology reduced barriers to enter the financial services markets, which has resulted in a wave of newcomers. Ecosystems are developing, allowing users to access social media, online purchases, services and payments all in one app. We believe open platforms and digital ecosystems will define the customer experience of the future, also for financial services. We are in the midst of the digital transformation reinventing ourselves.

Use case:
Software refactoring, or the act of improving the internal structure of a software system, is a fundamental practice for the sustainable maintenance of the fast-paced and long-lived software systems that all sectors of society develop.
Not paying attention to the internal structure of the system may incur high costs. The Consortium for Information and Software Quality estimates that, in 2018, the USA spent around US$ 2.84 trillion due to poor software quality, where around 18% (or US$510 billion) due to technical debt (i.e., bad technical decisions that developers took throughout the development) and 21% due to challenges in legacy systems (where, more often than not, we observe that the bad decisions from the past prevent developers from evolving the software).
Refactoring work in ING is also challenging. At ING, we have a self-build and maintained application with 20+ years of history and 4 mln lines of code and a combination of Java, PLSQL. This tool is used for collecting and distributing data so ING does have a correct and very detailed insight in our outstanding risk on our Lending book. We need to not only make sure keep the same logic, but also improve the performance by at least 5 times. Of course, code quality and security should be ensured.
Developments in the area of generative artificial intelligence (GenAI) are coming thick and fast, as we're seeing with the growing use of applications like ChatGPT in everything from search engines to composing music. ING, as other banks, is also looking at ways we can use GenAI to make the difference for our business and our customers, as well as our engineers.

We want to explore how the GenAI based technology such as Github copilot, ChatGPT, or other AI based technologies like transformers, https://huggingface.co/docs/transformers/index could help the manual refactoring procedure, like understanding the code, code review, code generation etc.

Expected project outcome
The expected project outcome is to find out which areas Gen AI technology could help during software refactoring, what tools can be useful, and how effective it will be to use Gen AI. The research questions to be answered are:

- How could AI facilitate the engineering work on software refactoring?
- How could AI help on understanding legacy software?

We would like to have two teams to explore these research questions separately. We could also provide two different use cases from ING (with the code from open-source project) to support the teams on this.

Technologies involved
The following technologies might be involved in this project (but not limited to):

- Git and Github
- PL/SQL, COBOL
- JVM technology
- Gen AI

Case: Contractuo

Title: How to change a trained transformer's mind?

Company: Contractuo

We happily call ourselves 'the least sexy startup in the world'. Why? Maybe some background will help! We started almost 4 years ago in the middle of the very first corona lockdown with a single mission: "enabling frictionless administrative processes through easy to use-, standardized, intelligent tooling…… so that companies can focus on their core business". In other words: we built the most boring tool for companies to smoothen all of their administrative processes so that they can focus on innovating in their own field! The solutions we provide let our customers work with data and automate data.

However, just because our solution might not be the sexiest doesn't mean it does not solve real problems! Since starting, we have been able to get some of Europe's biggest fund- & asset managers as customers as well as the worlds' biggest automotive parts supplier and we continue to convince companies leading in their respective field.

Use case:

Why Language Models

One of our objectives is to make the work of our users as easy as possible. By specifically training & implementing multiple models we can not only let them quickly assess risks but also automatically create tasks from the most important data, see what their legal department has to say and many other quality of life features.

The challenge

Can we get a transformer based model to get to 'automatically' learn new rules based on text by guiding it along several structured learning approaches.

Imagine that a household or a company has written policy. This policy stipulates things that you must do, things you are not allowed to do, things you may do and things to generally avoid. Seeing the nature of regulation, highly changing, it would make sense for central experts to be able to quickly 'teach' a model these new rules by following a couple of structured approaches to writing the rules. This would in turn help everyone to always stay up to date and even get instant feedback during process, creation of document and within context.