

Technische Universität Berlin
Fakultät IV - Elektrotechnik und Informatik
Fachgebiet AOT
Prof. Dr. Sahin Albayrak

Bachelorthesis proposal

Sparked, an intuitive user interface for the
automated machine learning project CODA

Robin Ruth
Matrikel-Nummer 316672

Betreuer Researcher Christian Geissler Dipl.-Inform. ABC

Contents

1	Motivation	2
2	Objective	3
3	Work packages	5
4	Schedule	6
5	Organizational	7
6	Appendix	8

1 Motivation

With the project CODA GT-ARC is doing fundamental research in automisation of selecting algorithms and hyperparameteroptimisation in the creation of of solutions in machine learning (taken from CODA Projektsteckbrief ... REMINDER: create bibentry).

Sparked builds on top of that, trying to give machine learning specialists and enthusiasts a clear and easy to use interface to interact with the current and future versions of the CODA project. Allowing the user to upload their own data, selecting out of the ever growing number of evaluation methods and metrics, to pick a classifier and select its parameters or request CODA to find the best parameters automatically and display the results in a visually appealing way, that allows for easy visibility of the wanted information.

With this Sparked becomes the face for the research done in CODA, to quickly demonstrate its capabilities or allow the informed user to test their own.

2 Objective

With Sparked we want to create allow the user to access the full range of possibilities that the CODA project offers in an easy to use, understandable manner.

CODA allows to calculate orders based on parameterised classifiers, evaluation methods, metrics and a set of testdata. These orders are a set of tasks, where a task is the unit that CODA can work with. Orders may have several parameterised classifiers, which will then be split into tasks with one classifier each. (NOTE: I believe we had defined, that dataset and eval method are fixed, but I can't quite remember if metrics was fixed too.) Sparked needs an interface to create those orders from a variable pool of classifiers, evaluation methods, metrics and datasets and run the resulting tasks against a given CODA backend.

Additionally a Sparked user might want to upload their own testdata. These csv files need to be uploaded to a selectable topic on a kafka instance, where the data is then picked up and transformed and saved to an hadoop filesystem.

Since the CODA backend may at this time only calculate one task at a time, tasks must be saved and send to CODA whenever there is free capacity. These saved tasks can have 4 states, finished, running, waiting and paused. The user should be able to see all created orders and tasks with there status, pause and stop orders, see the progress of the running task and open the analysis page for finished tasks.

In the second part of this project we want to evaluate if the goal of easy usability has been reached.

For this we will give the finished application to a machine learning expert, that has not previously seen the interface, allow him or her a limited time to use the program without help and then interview them on their perception of the usability.

Additionally we will discuss the used solutions and compare them to user interfaces of similar programs.

OLD Notes, V0.1:

- visually appealing [very oppinion based]
- display interesting data in an easy to find way [very oppinion maybe a list of information that is of interest for the user and an evaluation how many clicks/scrolls they need to view them? Thats not at all precise...]
- ease of use for ml experts [interview with expert?!]
- supports a demonstration mode, where the user starts with a clean slate (programming)

- allows the discussed functionality (programming, finishing specification)
- stable version [load test]
- configurability (changes in the known list of changeable objects (kafka server, spark server, kafka topics, classifiers, validation methods, evaluation methods, datasets))
- changable (code designed in a way to make changes easier)

3 Work packages

- Evaluating technologies (mostly done) 3 of 3.5 d
- Technical setup (partially done) still left: 1 of 5,5 d (get the kafka connection running, get everything running on docker, link docker images, move git repo to the gt-arc repo, make sure everything runs on the linux vms it needs to run)
- Proposal (started) 0.5 of 3 d + communication time
- Evaluation state of the art () 1.5d
- Programming (started) 5 of 25 d
- Bugfixing (this will happen and must be considered) 4d
- Creating expert interview (Questionnaire, Showcase, Interview, Evaluation) 6T
- Writing thesis () 20d (which is a bit below the 50 percent that you said I should reserve for this.)
- Feedback and corrections 6d
- Presentation () 2d (1,5d preparation, 0,5d the presentation itself)

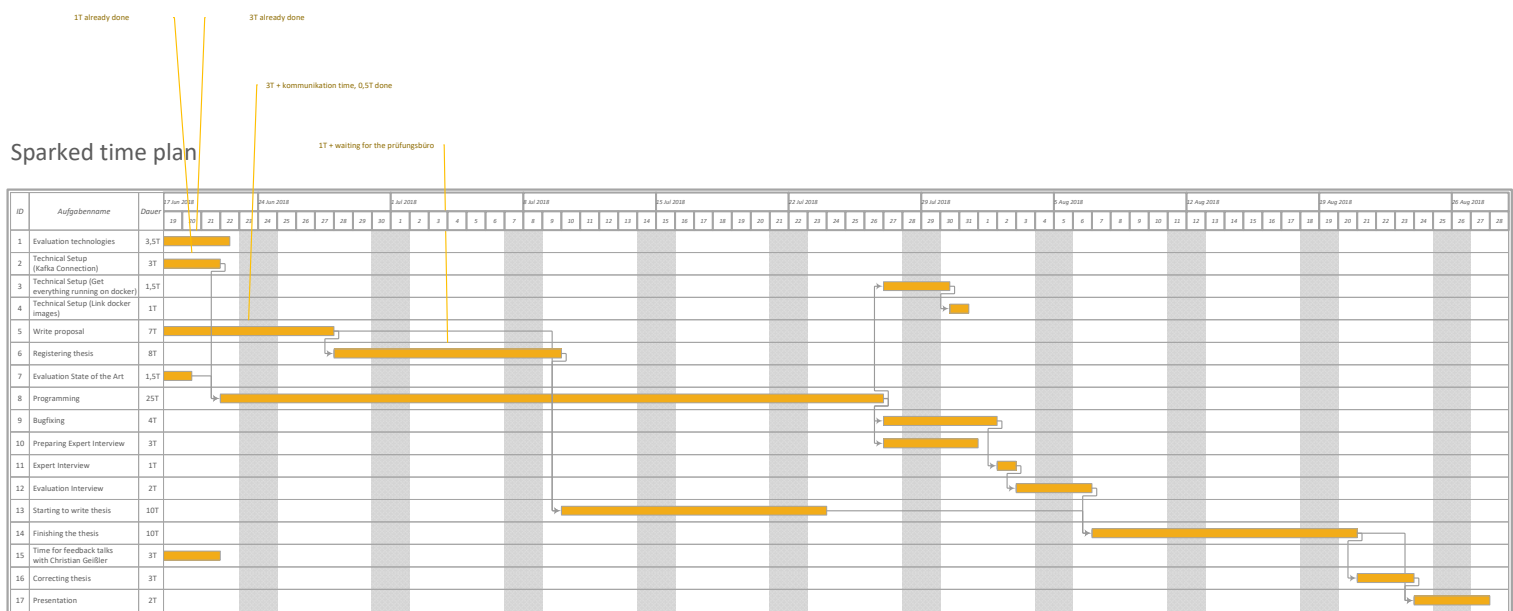
I am assuming 8h days.

My problem here is, that I have by my estimation about 76,5 days worth of work, plus everything I have yet forgotten and there is no real time for anything unforeseen in the plan. If that is somewhat accurate then, in the best case scenario, this is about 70 percent more time then the 12LP that the Prüfungsordnung says it should take.

4 Schedule

The time allotted to a bachelor thesis as defined in my examination regulation (StuPO-BSc-INF-2014) is 20 weeks with an time expenditure of 12LP.

The Gantt diagram shows the known work packages with their dependencies. See also Appendix.



5 Organizational

- Language of this Bachelorthesis is english.
- The thesis will be written with pdf_latex.
- The programming of the user interface uses several languages and technologies.
 - Java with Spring MVC for the backend
 - Angular 5 (typescript) and css for the frontend
 - MongoDB for data persistence
 - chart.js to display results in charts
 - Jackson 2 for automated conversion of data between JSON files and Java objects
 - Docker on linux as hosting system for all used systems
 - Postman for api evaluation
- Supervisors is Christian Geißler
- Evaluators are Prof. Dr. Albayrak and Prof. Kao
-

6 Appendix

3T already done

- 3T + kommunikation time, 0.5T done

1T + waiting for the prüfungsbüro

Sparked time plan

