

Software Engineering Assignment

Dennis Wilkman

June 2025

1 Research Introduction

The topic of my thesis is 'Learning for Dynamical Systems', where Learning is basically machine learning, and Dynamical Systems are any systems which change with time. In essence the aim is to merge machine learning with automatic control. A dynamical systems typically looks like this:

$$\frac{\partial}{\partial t}x(t) = \dot{x}(t) = \mathcal{F}(t, x)$$
$$x(0) = x_0, t \in \Omega_t, x \in \Omega_x$$

Where x is the state of the system and $\mathcal{F}(\cdot, \cdot)$ is a functional. And typically we use machine learning to learn a solution to the $x(t)$ of the system. One way to do it is to use physics informed neural networks. Which simply adds the dynamics of the system as a loss which we can sample across the space of interest. This simple way of combining dynamical systems and machine learning also for faster computations of solutions to the dynamical systems compared to numerical methods at the expense of accuracy, which is desired in certain situations.

2 Lecture Principles

2.1 Software Testing

Software testing is in my opinion quintessential to projects involving several people, since when one person does changes, they might not consider how it affects code that another person has written, as such it is very easy to accidentally break something unintentionally. Although not guaranteed, having automated testing of the code and its various modules will catch a lot of issues before they are merged into the code data base, and will save time in testing the modules every time new code is created or old code is changed. The downside is that sometimes the tests can be very time consuming to create and from my experience, sometimes when every single minor module to the code is expected to have some form of test attached to it, a lot of the tests done can feel very

unnecessary and spending 1 hour on creating tests for a module that took 5 minutes to create is in some ways very inefficient.

Although in terms of research topics, since I do not collaborate with anyone and I am not creating larger code bases/developing any products, actually using any form of software testing is in my opinion not deemed necessary. However, it is not unrealistic for the methods I am research on to be included in public libraries, if that would be the case tests would be necessary to ensure quality and bug free methods.

2.2 Behavioral Software Engineering

Although not directly related, I think there are some considerations in regard with how people in the field the reliability of the methods I propose. Typically in controlling dynamical systems guarantees are very important, and due to the stochasticity in neural networks, providing assurance that the methods are reliable is very important for the adaptation of them in real world applications. Therefore it is very important to learn how people working with dynamical systems view safety, and adjust my messaging accordingly to convince them of the safety and efficacy of my methods.

3 Guest-Lecture Principles

3.1 Understand the problem before building the solution

When conducting research and actually moving it to real world applications it is important to understand what specific real world problems we want to solve. Perhaps theoretically it could be of great importance, but in the real world, it is just not worth the hassle to switch methods just for a minute improvement. Therefore actually getting an idea of areas where it would be worth the hassle to make the switch and targeting the development of the software for those areas, could be a viable direction of development.

3.2 Goal Modeling

Although arguably in the same direction as understanding the problem before building the solution, when actually goal modeling try to have a clear idea of the direction and the problems we aim to solve with the implementation of our methods. Also understanding the competing methods already existing to understand the gap we are actually trying to fill and what we need to achieve to properly fill it.

4 Data Scientists versus Software Engineers

4.1 Difference between the roles

Not really much to add, I just agree with what they put forward. From my experience as a data scientist working closely with software engineers the skillsets they gave of the various roles aligned with what I have seen in practice.

4.2 Specialization vs Merging

Merging the roles is just one big conspiracy, management just wants the roles to merge so they can underpay one person to do a two man job. I do not think they should be the same roles, the skills required to be a data scientist (creating the models and handling the data) is quite different from actually moving the models to production. I believe a software engineer should be responsible to handle all the issues of moving the model to production while working with the data scientist to handle concerns that might be specific to machine learning models in production.

5 Paper analysis

5.1 Paper 1

The first paper I read was "How Do Model Export Formats Impact the Development of ML-Enabled Systems? A Case Study on Model Integration" (<https://arxiv.org/pdf/2502.00429>)

5.1.1 Core ideas and SE importance

In order to use any machine learning model (given that you do not have a computer running 24/7 with no backups of the model so on and so on...) you need to export the model in some way. Which in essence means saving the model somewhere as a file. Then later when you actually want to use the model wherever, you just load the file and infer (or do what you aim to do with it). The importance of this paper is to give AI-engineers an idea of what export format suits them the best depending on the model they are using and the tech stack they are utilizing. By answering three questions for the different export format candidates: How effectively can the export formats be integrated in software systems; By what extent is the integration affected by the scale and complexity of the machine learning models used?; How much technical documentation and support exists on the web?

5.1.2 Relation to my research

In my research, if I were to pre train a model for serving it would come in handy to use the findings of the paper to choose which export format would suite me

the best. However currently I believe the research I am currently doing does not actually reuse a trained model, instead a model is trained, inferred once, and then thrown away. As such model exportation does not actually fit the scheme, instead should the method be deployed on a server, data would be received, a model be trained, inferring on the points of interest, returning the inference and discarding the currently trained model.

5.1.3 Larger AI intensive project integration

I would say almost any real life machine learning product development use case. For example, lets say you train a model which does supervised learning and does classification, in order to deploy the trained model, the model needs to be exported to then be deployed somewhere ready for inference. Unless the model is trained inferred and discarded as my previous case. It would be ideal to just export the model to reuse wherever needed.

5.1.4 Adaptation of your research

The paper suggested that the ONNX framework works quite well, as such, if there were any models I would like to deploy and required a framework to export any machine learning model I trained I would use it. Although when that day comes I would also verify that no new superior frameworks have appeared.

5.2 Paper 2

For the second paper I read "LoCoML: A Framework for Real-World ML Inference Pipelines" (<https://arxiv.org/pdf/2501.14165>)

5.2.1 Core ideas and SE importance

LoCoML addresses the actual integration itself into software engineering systems compared to the previous paper which only addressed the model exportation itself. I.E. in this scenario they would ideally use the works of the previous paper to choose the correct format to then use for deployment in their systems. After all the models have been trained/saved (the first part of the framework) they suggest a "Pipeline Orchestrator" consists of a "Pipeline Designer" which in turn uses a "Pipeline Builder" to construct pipelines sequentially which could include model training, data-preprocessing and more! The validation is managed by a "Pipeline Validator" which is also a component of the "Pipeline Builder". This in turn gives a general framework in which machine learning models are prepared for deployment in various systems.

5.2.2 Relation to my research

Similarly to the previous paper, it only uses saved models (even if they include training it, it is only then saved to be reused later, which in my use case would be redundant). Therefore as previously mentioned it would only be relevant if

I would do research on models which are actually saved. Even then this is an even more usecase, the previous one exportation of a model is always done when saving, the scenarios in which this method would be useful is only for certain ML integrations into software.

5.2.3 Larger AI intensive project integration

The example they give is related to this, a project in which you aim to integrate various ML technologies such as automatic speech recognition, machine translation and text to speech. A similar example could be integrating technologies such as audio classification, image classification to classify things in a multi modal setting.

5.2.4 Adaptation of your research

I can use the papers ideas to create deployment and inference pipelines which follow the guidelines of the paper. Although I believe in my potential use cases this is severe over engineering, I do believe there are some aspects I can draw inspiration from, like the heavy modularization. Also if I ever where to create a model to be deployed, I could get some idea of how it would actually be deployed in practice, and as such adjust it to more seamlessly fit the potential usages.

6 Research Ethics and Synthesis Reflection

6.1 Search and Screening Process

I found the conference, looked at some papers that seemed interesting, they were red hearings, I looked at some more and found ones that seemed relevant. Also since for some reason I did not manage to get access via my university I found the papers on arxiv to read.

6.2 Pitfalls and Mitigations

In the beginning I encountered papers that where more focused on how to use the model than the software engineering side. To counteract this I focused on titles which really seemed to be about software engineering (or atleast fits my definition of software engineering).

6.3 Ethical Considerations

What is this even for a question? I am not using an LLM and I am not copying someone else work, I ensured originality by just not stealing others peoples works and not using an LLM to help me. Although a lot of originality stems from me yapping about my personal experiences with the subjects.