Agenda

• 13:00: Welcome: Anne Katrine Windfeld, Grundfos
• 13:10: AI: from experimentation to delivering value: Jan Bosch, Chalmers University of Technology & Software Center
• 13.30 Trustworthiness in testing and continuous integration: Kristian Sandahl, Linköping University
• 13.50: Presentation of selected Software Center projects: Helena Holmström Olsson, Malmö University
  - Accelerating Digitalization Through Data
  - Strategic Ecosystem-Driven R&D Management
• 14.15: Interactive workshop on digitalization
• 15:45: Closing: Jan Bosch
Artificial Intelligence
from experimentation to delivering value

Jan Bosch

Director Software Center
www.software-center.se
Professor of Software Engineering
Chalmers University of Technology
Gothenburg, Sweden.
Overview

- Vem är jag? Wie ben ik? Who am I?
- Reflecting on AI
- AI Engineering
- Example research activities
- Conclusion
Software Center

Mission: To significantly improve the **digitalization** capability of the European Software-Intensive industry
Some Online Companies

- Booking.com
- Spotify
- Rovio
- Klarna
- King
- Microsoft
Advantages for Industry

• More consistent, integrated focus on your key change initiatives
• Holistic approach including technical, organizational and business aspects
• Value every 6 months
• Opportunity to steer projects frequently
Three Key Take-Aways

- Digital companies need to be world-class at software, data and AI
- Moving AI from prototyping and experimentation to deployment in production requires significant engineering effort
- AI engineering is concerned with software engineering for AI
Overview

- Vem är jag? Wie ben ik? Who am I?
- Reflecting on AI
- AI Engineering
- Example research activities
- Conclusion
The new spring in artificial intelligence is the most significant development in computing in my lifetime.

— Sergey Brin, President ALPHABET
Artificial Intelligence
Why Machine Learning Revolution now?

1. INCREASE OF TRAINING DATA
   - Large datasets of labeled training data like ImageNet

2. INCREASE OF COMPUTING POWER
   - Especially GPUs & Distributed Computing.

3. ALGORITHMIC IMPROVEMENTS
   - Especially in deep reinforcement learning but also normal DL (RelUs [Dahl et al., 2013], Dropout [Srivastava et al., 2014], etc.)

4. BETTER TOOLING
   - Libraries like TensorFlow and Cloud Services
The Typical Problem

50%-85% of AI projects fail\(^1\)

\(^1\)https://www.pacteraedge.com/pactera-white-paper-reveals-85-percent-ai-projects-ultimately-fail-0
If you haven’t failed, you’re not trying hard enough.

Jennifer Crusie
Why Software Engineering For AI?

Where The Effort Goes ...

assemble datasets → create models → train and evaluate → deployment

Amount of effort  

Amount of attention
“One of the biggest problems facing businesses is getting AI from pilot phase to scalable deployment.”

KPMG Consulting
Five Most Important Jobs in AI

AI Architect – These specialists look at individual business processes - as well as the big picture organization - and determine what they can inject and embed AI successfully. They are also responsible for measuring performance, and sustaining the AI model over time - ensuring it removes mundane tasks to optimize humans in the workforce. The lack of AI architects is a big reason why companies cannot successfully sustain AI initiatives.

AI Product Manager – Working closely with the development team, an AI product manager serves as a liaison across multiple business teams to ensure AI models are implemented. They also work closely with these teams – as well as HR – to identify and integrate AI technology to ensure optimal performance of both humans and machines.

Data Scientist – With the ever-growing amount of data available to businesses, there is a shortage of experts with the skills to clean this data, and then design and apply the appropriate algorithms to glean meaningful insights.

Software Engineer – One of the biggest problems facing businesses is getting AI from pilot phase to scalable deployment. Software engineers work hand-in-hand with data scientists to bring AI into production, blending business acumen with a deep understanding of how AI works.

AI Ethicist – As ethical and social implications of AI continue to unfold, companies may need to create new jobs tasked with the critical responsibility of establishing AI frameworks that uphold company standards and codes of ethics. Initially, these roles could be fulfilled by existing leaders in an organization, but as the effects of AI truly take shape, it may need to be the responsibility of one person to ensure these guidelines are upheld.
AI: The New Electricity
Holistic DevOps Framework

Requirements driven development
- Regulatory features
- Competitor parity features
- Commodity features

Outcome/data driven development
- Value hypothesis
- New “flow” features
- Innovation

AI driven development
- Minimize prediction errors
- Many points in data set
- Combinatorial explosion of alternatives

Holistic DevOps Framework

AI component
SW component
continuous deployment
behavior data
System in operation
In God we trust; all others bring data.

W. Edwards Deming
Overview

• Vem är jag? Wie ben ik? Who am I?
• Reflecting on AI
  • AI Engineering
• Example research activities
• Conclusion
AI Engineering

• AI engineering is concerned with applying (software) engineering principles to the creation, deployment and evolution of AI driven systems.

• AI engineering focuses on the business, architectural, process and organizational implications of using ML/DL models in industrial systems, including data pipelines, monitoring & logging, quality attributes, etc.
How AI Evolves In Industry

- Experimentation & prototyping
- Non-critical deployment of ML/DL components
- Critical deployment of ML/DL components
- Cascading deployment of ML/DL components
- Autonomous ML/DL components

Increasing need for AI engineering
AI Engineering Research Agenda

Overview

- Vem är jag? Wie ben ik? Who am I?
- Reflecting on AI
- AI Engineering
- Example research activities
- Conclusion
<table>
<thead>
<tr>
<th>Study #2</th>
<th>Experiment Prototyping</th>
<th>Non-critical deployment</th>
<th>Critical deployment</th>
<th>Cascading deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>assemble dataset</td>
<td>Issues with problem formulation and specifying desired outcome</td>
<td>Data silos, scarcity of labelled data, imbalanced training set</td>
<td>Limitations in techniques for gathering training data from large-scale, non-stationary data streams</td>
<td>Complex and effects of data dependencies</td>
</tr>
<tr>
<td>create model</td>
<td>Use of non-representative dataset, data drifts</td>
<td>No critical analysis of training data</td>
<td>Difficulties in building highly scalable ML/DL pipeline</td>
<td>Entanglements causing difficulties in isolating improvements</td>
</tr>
<tr>
<td>train and evaluate model</td>
<td>Lack of well-established ground truth</td>
<td>No evaluation of models with business-centric measures</td>
<td>Difficulties in reproducing models, results and debugging DL models</td>
<td>Need of techniques for sliced analysis in final model</td>
</tr>
<tr>
<td>deploy model</td>
<td>No deployment mechanism</td>
<td>Training-serving skew</td>
<td>Adhering to stringent serving requirements e.g., of latency, throughput</td>
<td>Hidden feedback-loops and undeclared consumers of the models</td>
</tr>
</tbody>
</table>

Table 2. Challenges in the evolution of use of ML/DL components in software-intensive systems

## Study #3

**Mapping between Data Management Challenges and Use Cases**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Challenge</th>
<th>RS(^1)</th>
<th>WPP(^2)</th>
<th>HPP(^3)</th>
<th>MD(^4)</th>
<th>FFD(^5)</th>
<th>MS(^6)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Collection</strong></td>
<td>Lack of metadata</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Data Granularity</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Shortage of diverse samples</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Need for sharing and tracking techniques</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data Storage</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Data Exploration</strong></td>
<td>Statistical Understanding</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deduplication Complexity</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Heterogeneity in data</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Data Preprocessing</strong></td>
<td>Dirty data</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Managing sequences in data</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Managing categorical data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Dataset Preparation</strong></td>
<td>Data Dependency</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data Quality</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Data Testing</strong></td>
<td>Tooling</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Expensive Testing</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Deployment</strong></td>
<td>Data Extraction Methods</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overfitting</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Post Deployment</strong></td>
<td>Data sources and Distribution</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data drifts</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Feedback loops</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Recommender System  
2. Wind Power Prediction  
3. House Price Prediction  
4. Melanoma Detection  
5. Financial Fraud Detection  
6. Manufacturing Systems

Conceptual Model for Data Pipelines

Data Generation → Data Collection → Data Ingestion → Data Extraction → Data Labeling

- Raw data storage
- Model Serving
- Model Evaluation
- ML/DL Modelling

- Sending alarm
- Authentication
- Validation
- Fault detection
- Mitigation

Chalmers University of Technology
Automatic data validation

- Irrespective of ML algorithms, data errors (e.g., training-serving skew) can adversely affect the quality of generated model

Need to ensure data errors are caught early in the context of continuous and frequent ML model training and serving
Automatic data validation

Define checks (from data processing and model training stages to data collection stage)

- Feature: Size, Missing Values, Unique Values, Range Values, Compliance of Values, Outliers, Malformats
- Dataset: Cross-Dataset: Missing Features, Distribution skew
- Datastream: Provide Feedback and Mitigation Strategy

At Cell
- Warn, report and monitor e.g., ratio/number of missing values
- Retrieve failed records and perform mitigation strategy: ignore, remove, impute missing values

At Feature
- Warn, report and monitor e.g., ratio/number of duplicates per feature
- Retrieve failed records and perform mitigation strategy: ignore, remove or correct duplicates

At Dataset
- Warn, report and monitor e.g., missing features in subsequent dataset
- Retrieve dataset with deviations and perform mitigation strategy: ignore, remove or add missing features

At Feature and Dataset
- Warn and monitor e.g., distribution skew
- Transform or update data collection procedure
- Mark validation error as data is collected

Legend
- Check or constraint

Transferable check or constraint

Feedback
- Mitigation strategy
Automatic Labeling

Accuracy (all datasets)

Value

Model
- LabelPropagationKNN
- LabelPropagationRBF
- LabelSpreadingKNN
- LabelSpreadingRBF
- QBC
- RandomSampling
- UncertaintySampling

HPDI of the strength of the model

Model
- UncertaintySampling
- RandomSampling
- QBC
- LabelSpreadingRBF
- LabelSpreadingKNN
- LabelPropagationRBF
- LabelPropagationKNN

Estimate
Deep Semi-Supervised Learning

The two most common learning paradigms are Supervised learning and Unsupervised learning.

Supervised learning utilizes labelled data.

The more labelled data you have the better, especially for deep learning algorithms.

In industrial settings labelled data is often scarce.

Unsupervised learning utilizes unlabelled data

Learns patterns from unlabelled data.

Unlabelled data is easy to find

Question: Can we learn from both labelled and unlabelled data and can this improve supervised classification?

Yes we can!

Semi-Supervised Learning
Automated Experimentation
Multi-armed bandit algorithms
Federated Learning

(a) Traditional Machine Learning system

(b) Federated Learning system
Aynchronous Federated Learning
Decentralized Reinforcement Learning - System Scenario

Fig: UAV-assisted system design: Mission-critical scenario enabled by three wireless drones
Decentralized Reinforcement Learning

The system has two different channels for exchanging information:

- The brown line indicates the sensor data channel, which will communicate to nearby drones connection performance and UAV-BS location data.

- The local model of each UAV-BS will be shared with its neighbours via the model data channel, which is indicated by the green line.

- Each UAV-BS will share their learning experiences as a result, and the others can gain information from the experiences of the others. After multiple training epochs, the UAV-BS can swap their model with their neighbours under the control of a frequency parameter.
To assist companies to determine whether to deploy AI models in the cloud or on the edge, we evaluate architectural alternatives for deploying AI on the edge based on the key factors.

- Involves experimentation with 3 architectural alternatives:
  - Centralized architecture
  - Federated architecture
  - Decentralized architecture

- Validate these architectures against key factors
  - **Mass customization** - Extent to which a local model can be customized and how it impacts overall architecture
  - **Scalability** – Measures how effectively the local models scale from one-to-many instances
  - **Model Performance** – Indicates the overall quality of the architecture

- Implementation:
  - Using SNIC (2 servers and 3 edge instances)
  - Three datasets - MNIST, CIFAR-10, ECMWF
  - Model - CNN
Why monitoring is important??

- To track the performance of ML/DL models in production and ensure that business objectives are met, model monitoring is a key step
- Even though ML/DL practitioners may benefit from existing lessons and approaches from software monitoring, ML/DL model monitoring presents unique challenges that necessitate the use of specialized techniques, processes, and tools
- To address these challenges, we propose to conduct in-depth studies focusing on the specifics of monitoring models after they have been operationalized in embedded systems.

Research Questions:

RQ. How to monitor models to ensure that it performs well once it is operational?
   
   RQ1. In software-intensive embedded systems companies, what are the related activities and challenges that practitioners experience while monitoring ML/DL models and what are the potential solutions to these challenges?
   
   RQ2. What are the different ways to compare results from a newer version(s) of the model against the in-production version(s)?
   
   RQ3. How to build an infrastructure around the ML/DL model for monitoring to make sure it performs well once it is operational?
   
   RQ4. Once the model has been deployed to different clients in the field, how can we identify subgroups of clients with similar characteristics that underperform?

Methodology:

- Conduct the research either on-site or online --- If on-site, we propose action research (work in close collaboration with experts in the company), Otherwise, in-depth case study.
- The overall research involves interviews, workshops, focus groups, prototyping/demonstration, experimentation and validation.

For more information, please contact meenu-mary.john@mau.se, helena.holmstrom.olsson@mau.se or jan.bosch@chalmers.se
Overview

• Vem är jag? Wie ben ik? Who am I?
• Reflecting on AI
• AI Engineering
• Example research activities

• Conclusion
Artificial intelligence is a tool, not a threat.

Rodney Brooks
Conclusion

• Digital companies need to be world-class at software, data and AI

• Moving AI from prototyping and experimentation to deployment in production requires significant engineering effort

• AI engineering is concerned with software engineering for AI
Learn More?

- Speed, Data, and Ecosystems
  - Excelling in a Software-Driven World
  - Jan Bosch

- Using Data to Build Better Products
  - A Hands-On Guide to Working with Data in R&D - The Basics
  - Jan Bosch

- Impactful Software
  - Jan Bosch

- Digital Transformation
  - A Holistic Perspective for Business Leaders
  - Jan Bosch
Follow me on LinkedIn, Twitter (@JanBosch) or
www.janbosch.com/blog
Closing thoughts

• Professional development
• Echo chamber
• Role
• Thank you!
  Grundfos for hosting
  DEIF and Grundfos for being members
  All of you for participating!
• What’s next?
  Software Center reporting workshop – December 15 (and June 15 next year?)
  Join Software Center! And if you’re a member, join a(nother) project!