Ivica Crnkovic
Chalmers University of Technology
Professor in Software Engineering
Director of Chalmers AI Research Centre (CHAIR)
ivica.crnkovic@chalmers.se
Types of Machine Learning

SUPERVISED
- Task Driven
  - predict next value, classification
- 95% of ML

UNSUPERVISED
- Data Driven
  - Identify clusters

REINFORCEMENT
- Learn From Mistakes
  - Trial and Error
AI as effective as medical specialists at diagnosing diseases.

AI in transportation:
• Self-driving vehicles, Traffic management, delay predictions...

AI and NLP (Natural Language processes)

Traffic, Medicine, Robotics, Finance, Health, Space

AI success Stories

virtually no area that does not (plan to) use AI
AI failures 2018

Amazon AI recruiting tool is gender biased
AI failures 2018
AI World Cup 2018 - predictions almost all wrong 😊
Why these failures?

What are the challenges in AI development?
New questions in AI Development

• **Non technical character**
  • Who is owner of data?
  • What are the ethical aspects of using data?
  • What can we allow a machine to decide?
  • How do interpret the results from AI models?

• **Technical nature**
  • How to efficiently collect, store, process, analyse, and present data?
  • How to efficiently build the AI-based systems?
  • How to ensure enough resources (computation, storage, timing)
  • How to ensure dependability/trustworthy of such systems?
  • What system and software architecture are required for AI-systems?
  • **WHAT KIND OF SOFTWARE ENGINEERING SUPPORT IS NEEDED?**
ML Development cycle

1. Assemble datasets
2. Create models
3. Train and evaluate
4. Deployment

The cycle is iterative, moving from assembling datasets, creating models, training and evaluating them, and then deploying the model before returning to the start.
Life cycle

Example: vehicular system (a Cyber-physical system)

- Collecting data
- Analyzing Data
- Controlling system

System evolution

Updating system

Sending data

https://www.youtube.com/watch?v=XEHG2oYdsiU
Architecture of a car control system
Complex services – distributed components

Challenges:
- Real-time requirements
- Shared resources
- Resource constraints
Software architecture

Subsystem and components

Component-based and service-based approach
- Components
  - Encapsulation of data
  - Encapsulation of functionality
  - Dependency between components defined and controlled
### AI-System - system architecture (example)

<table>
<thead>
<tr>
<th>Subsystem A</th>
<th>Subsystem B</th>
<th>Subsystem C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Middleware</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Platform</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HW</td>
</tr>
</tbody>
</table>

Components – black boxes
- Components
  - Encapsulation of AI-based functionality
  - Dependency between components defined and controlled
  - What about AI code and data?
AI-based vehicular system
- large amounts of data processing
- large computational requirements

Collecting more data
Analyzing more data
More computation
Control system
Further data processing

Updating system
Sending data
For diagnostic
And for ML

System evolution

Collecting data
System training

CLOUD COMPUTING

https://www.youtube.com/watch?v=XEHG2oYdsiU
With AI the existing architecture is not feasible
Challenges

- New computation models
- Heterogenous platform
- Edge sensor vs. edge centralized level vs. cloud computing

- Development process
  - separation of training process from operational process
  - New monitoring
  - New quality attributes and their metrics
  - Migration process from the existing architecture
AI-based components – fundamental problems

- The results depend not only on the algorithms and controlled data but also on uncontrolled/unknown data.

- The AI-based functions are not continuous: small change of data can cause big changes.

- Due to data impact on the model, many new challenges appear.
Examples of Data-related challenges

1. Entanglement (Data fusion)

2. Data dependencies
   - Unstable data dependencies
   - Underutilized Data Dependencies

3. Hidden Feedback Loops

4. Undeclared Consumers

5. Correction Cascades
System-design anti-patterns (1)

- **Heterogeneity of data** (different formats, accuracy, semantics) and use of standard ML functions require a lot of Glue code
  - 95% of code in AI-based systems is a glue-code *(empirical data)*
  - Requires
    - Frequent refactoring of code
    - Re-implementing AI models

- **Pipeline Jungles**
  - ML-friendly format data become a jungle of scrapes, joins, and sampling steps, intermediate files
    - Requires – a close team work of data and domain engineers
System-design anti-patterns (2)

Dead Experimental Code paths

- AI solution requires a lot of experimentation
- A lot of code that will not be used later
- Problems
  - Dead code

- Version management – how to preserve useful configuration branches, and remove unnecessary
Managing Changes in the External World

Continuous change of data

Challenge: models and system behaviour dependent of data
Examples:
- Threshold changes
- Correlation between data

Requirements: Continuous monitoring of data and system. Continuous test.
Study: Overview of industrial ML systems

Which challenges are the most difficult in development of AI-based systems?

Case studies

A. Project EST: Real Estate Valuation
B. Project OIL: Predicting Oil and Gas Recovery Potential
C. Project RET: Predicting User Retention
D. Project WEA: Weather Forecasting
E. Project CCF: Credit Card Fraud Detection
F. Project BOT: Poker Bot Identification
G. Project REC: Media Recommendations
Study: Engineering challenges of DL

Hard to solve

- Dependency Management
- Troubleshooting
- Testing
- Development challenges
- Production/operation challenges
- Organisational challenges

Effort estimation
- Glue code and supporting systems
- Resource Limitations
- Privacy and Safety/security
- Limited transparency
- Unintentional feedback loops
- Experimental management
- Monitoring and logging

Lower business impact

Easier to solve

High business impact
The challenges in evolution of development and use ML components

<table>
<thead>
<tr>
<th>Experiment prototyping</th>
<th>Non-critical deployment</th>
<th>Critical deployment</th>
<th>Cascading deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Assemble dataset</strong></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>
</tr>
<tr>
<td><strong>Create model</strong></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 pipeline</td>
</tr>
<tr>
<td><strong>Train and evaluate model</strong></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>
</tr>
<tr>
<td><strong>Deploy model</strong></td>
<td>No deployment mechanism</td>
<td>Training-serving skew</td>
<td>Adhering to stringent serving requirements e.g., of latency, throughput</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cases</th>
<th>identified problems</th>
<th>strategic focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Estate Valuation</td>
<td>Lack of labelled data</td>
<td>data quality management</td>
</tr>
<tr>
<td>Predicting Oil and Gas Recovery</td>
<td>Lack of metadata</td>
<td>design methods and processes</td>
</tr>
<tr>
<td>Predicting User Retention</td>
<td>Shortage of diverse samples</td>
<td>model performance</td>
</tr>
<tr>
<td>Weather Forecasting</td>
<td>Heterogeneity in data</td>
<td>deployment &amp; compliance</td>
</tr>
<tr>
<td>Credit Card Fraud Detection</td>
<td>Data granularity</td>
<td></td>
</tr>
<tr>
<td>Poker Bot Identification</td>
<td>Imbalanced data sets</td>
<td></td>
</tr>
<tr>
<td>Media Recommendations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor data (automotive)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentiment analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manufacturing optimization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training data annotation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Failure prediction (telecom)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OoO reply analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search engine optimization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wind power prediction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skin lesion classification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AI Engineering challenges

### Strategic focus
- Data quality management
- Design methods and processes
- Model performance
- Deployment and compliance

### Infrastructure
- Data versioning & dependency mgmt
- Federated learning infrastructure
- Storage and computing infrastructure
- Deployment infrastructure

### Development
- DataOps/DevOps
- Reuse of pre-developed ML models
- Quality attributes
- Integration of models and SW components

### Process
- Automated data labeling
- Manage multiple models
- A/B testing of models
- Monitoring & logging

---

Domain-specific AI Engineering challenges

Strategic focus
- Data quality management
- Design methods and processes
- Model performance
- Deployment and compliance

Cyber-physical systems
- Federated collection & storage of data
- Federated ML models
- Transfer learning
- Heterogenous HW platforms

Safety-critical systems
- Data trail
- Explainable models
- Safety validation
- reproducibility

Autonomously Improving systems
- Data generation for ML
- Automated experimentation
- Online evaluation
- Exploration vs. exploration

Example – Process integration

DevOps

ML Workflow

Data collection → Data preparation → Data labelling → Feature engineering → Model training → Model evaluation → Model deployment → Model monitoring

Data Management | ML Modeling | ML Operation

Lwakatare, Crnkovic & Bosch. DevOps for AI – Challenges in Development of AI-enabled applications, SoftCom 2020
Conceptual ML Workflow and DevOps Process Integration

Lwakatare, Crnkovic & Bosch. DevOps for AI – Challenges in Development of AI-enabled applications, SoftCom 2020
Excitement of a researcher

ML/DL introduces many new challenges for software

- On addition to software evolution there is data evolution
- Context change – in behaviour and in data
- New (distributed) system configuration
- Continuous integration, continuous deployment
- Continuous training

Many companies will not be able to cope with new types of complexity

Many opportunities for researchers!

Conclusion