Data Assurance in the Context of AI Pipelines

By: Evan Stosic

Overview

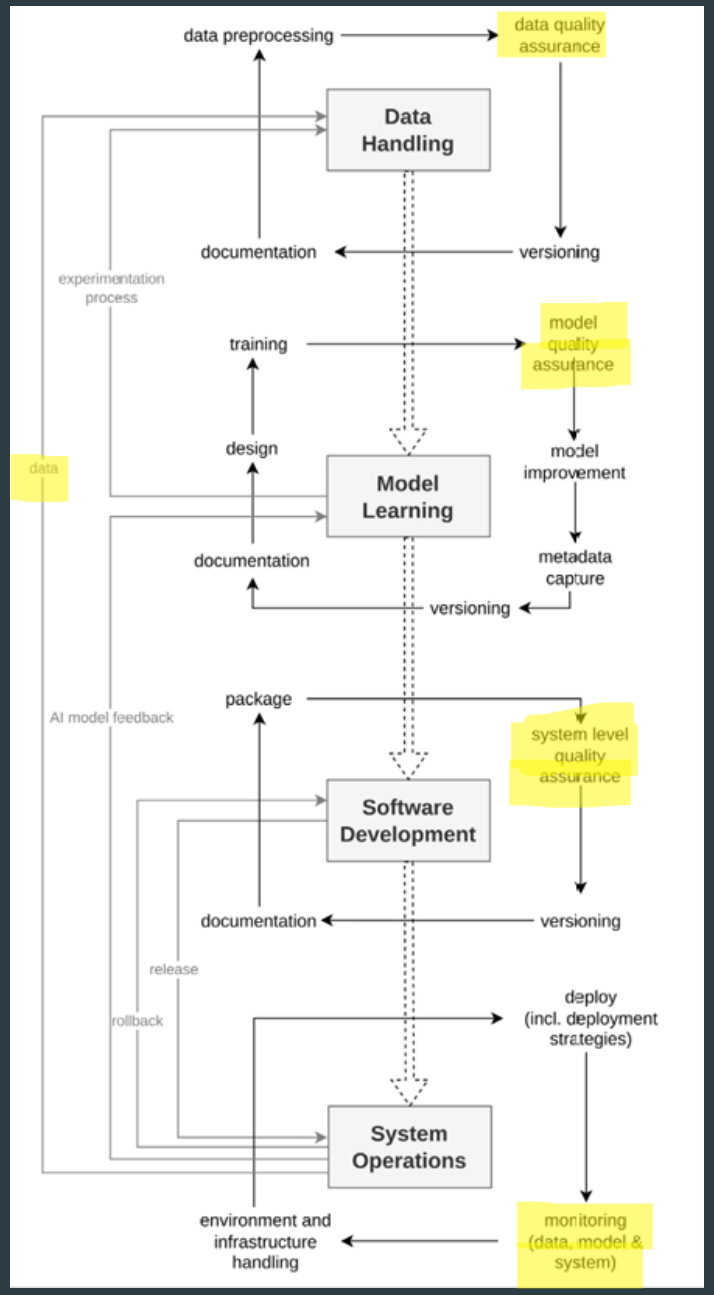
- AI Pipelines
- Data Assurance in AI Pipelines
- Example: Generative Classifiers for Image Processing

AI Pipelines

- To utilize AI models effectively, it is necessary to deploy and integrate AI models into production systems and to assure the quality of the resulting continuously evolving and self-adapting systems
- AI presents a unique problem, because of characteristics such as non-determinism (output is dependent on training data, unlike deterministic algorithms or systems where we have more control over the result)
- One solution: automated end-to-end CI/CD (continuous integration / continuous development) lifecycle pipelines
  - Well-established in traditional software development
  - Difficulties: Need to handle data, the AI model itself, and large system-level complexity

AI Pipelines

- 4 Pipeline Stages
  1. Data Handling
  2. Model Learning
  3. Software Development
  4. System Operations



Source: [1]

Data Assurance in AI Pipelines

- List of possible issues with data: bias, variance, incompleteness, data skewness, lack of structure, tampering by malicious parties
- These issues can happen at any point in the pipeline
- Most effective solutions seem to focus on a specific AI subarea and one domain
- Explainable AI (XAI): can identify how the outcomes of the model were arrived at
  - Makes traditional data assurance methods more applicable
  - Research on XAI in different areas (signal processing, remote sensing, computer vision)

Example: Generative Classifiers for Image Processing

- Discriminative classifiers (DCs): compute the probability of a class when given an input directly (class | image)
  - More widely used for their better performance on larger datasets
  - Difficult to model how we obtain an output
- Generative classifiers (GCs): compute the probability of an input image conditioned on each class (image | class)
  - Less used because for larger datasets, GCs produce more uncertain output
  - Individual output is more informative, e.g., can show if a prediction is uncertain because the input agrees with both classes, or with neither

Example: Generative Classifiers for Image Processing

- Approach: Train an INN (Invertible Neural Network) model using Information Bottleneck as a GC on the dataset
  - Information Bottleneck: trade-off between the complexity of representation and the power of predicting
  - Needed because classes have complex representations
- The following aspects of the resulting model were analyzed:
  - General Performance
  - Explainability
  - General Robustness
  - Handling Corrupted Images
  - Handling Adversarial Attacks

References

- [1]: https://www.sciencedirect.com/science/article/pii/S0164121223000109
- [2]: https://link.springer.com/article/10.1186/s40537-021-00445-7
- [3]: https://www.sciencedirect.com/science/article/pii/S277266222300070X
- [4]: https://arxiv.org/pdf/2007.15036
- [5]: https://ieeexplore.ieee.org/document/9131692