*International Journal of Latest Research in Engineering and Technology (IJLRET)*
*ISSN: 2454-5031*
*www.ijlret.com || Volume 08 - Issue 11 || November 2022 || PP. 24-30*

# Designing Frameworks for Reliability in Deep Learning Systems

Ayse Kok Arslan

**Abstract:** There has been a great amount of progress in deep learning models in the last decade. Such models are most accurate when applied to test data drawn from the same distribution as their training set. However, in practice, the data confronting models in real-world settings rarely match the training distribution.

This study explores the use of co-design approaches for developing reliable design frameworks for deep learning systems.It aims to raise awareness on how to develop reliable ML models within the context of recommender systems. While much work needs to be done in this field, the study providessuggestions and practical tips for how to develop reliable ML models such as in the case of recommender systems.

## Introduction

There has been a great amount of progress in deep learning models in the last decade. While the AI (artificial intelligence) communitystrives for providing openness and transparency within AI research, the inherent challenges of the field remain unchanged.One of the main issues is that when it comes to testing and experimentation, the data confronting models in real-world settings rarely match the training distribution. In addition to this, given the huge costs of running the models it becomes more challenging to make design judgements.

Given the plethora of these unresolved issues, this study aims to provide a review of co-design approaches for developing reliable design frameworks for deep learning systems. It starts with a brief overview of AI and co-design approaches. Next, it sheds light onto model architecture, especially for ranking and recommender models as the practical tips and suggestions relate to the context of a recommender platform. By doing so, it aims to raise awareness on how to develop reliable ML models.

Although the study provides an architecture and model recommendations specifically for a recommender system, the main points should be relevant for any other ML-driven model as well.

## Review of Existing Studies

Design involves making good judgments in pursuit of desirable design outcomes. Design judgments do not follow a formallinear or rule-based process,yet emerge depending on the designer's experiences and the contextual aspects of the design situation (Dunne, 1999; Nelson & Stolterman, 2014).

When it comes to making design judgments for machine learning (ML), the process can become more challenging. The generic definition of "an AI system" is a single unified software system that can reliably pass theadversarial Turing test in which the human judges are instructed to ask interesting and difficult questions, designed to advantage human participants, and to successfully unmask the computer as an impostor.

"Unified" means that the system is integrated enough that it can, for example, explain its reasoning on a Q&A task, or verbally report its progress and identify objects during model assembly.

It should also be noted that agents play a crucial role for any ML system.The central feature of agency is that an agent chooses a particular action because it "expects it" to delivera desirable certain outcomeagents are systems whose outputs are moved by reasons (Dennett, 1987). Main characterizations of agents include:

- An agent is a system whose behavior can be compressed with respect to an objective function (Orseau et al., 2018).
- "An optimizing system is ... a part of the universe [that] moves predictably towards a small set of target configurations" (Flint, 2020).
- A goal-directed system has self-awareness, planning, consequentialism, scale, coherence, and flexibility (Ngo, 2020).
- The intentional stance:An agent's behavior can be usefully understood as trying to optimize an objective (Dennett, 1987).
- Cybernetics:An agent's behavior adapts to achieve an objective (e.g. Ashby, 1956; Wiener, 1961).
- Decision theory / game theory / economics / AI: An agent selects a policy to optimize an objective.

Despite their agency, ML modelscan still become prone to adversarial attacks, therefore there is a need to determine how to prevent adversarial attacks and to develop reliable models. Nelson and Stolterman (2014) called this prioritization appreciative judgment.

As design is highly dependent on context, navigational judgment assists designers in adjusting their approach to the changing situational realities of a design situation. Designers use navigational judgment as a toolto deal with unpredictable situations such as adversarial attacks in ML models (Nelson & Stolterman, 2014).

When making design judgments on ML models, what would be needed to either complement or supplant example-driven trained neural network systems is the socalled common sense, which refers to the ability to see things that to many human-beings seem obvious, to draw fast and simple, obvious conclusions (Brachmann, 2005). To achieve this within a large number of domains, the model should learn many domain-specific sub-tasks (e.g., filtering different kinds of noise or focusing on a specific detail), which can only be learned from a semantically diverse dataset.

There are three reasons why model evaluation plays a crucial role:

1. To estimate the generalization accuracy, the predictive performance of a model on future (unseen) data.
2. To increase the predictive performance by tweaking the learning algorithm and selecting the best-performing model from a given hypothesis space.
3. To identify the machine learning algorithm that is best-suited for the problem at hand.

One of the probably most common technique for model evaluation is k-fold cross-validation. Here, the main idea behind cross-validation is that each sample in a dataset has the opportunity of being tested. k-fold cross-validation is a special case of cross-validation where one iterates over a dataset set k times. In each round, one splits the dataset into k parts: one part is used for validation, and the remaining $k - 1$ parts are merged into a training subset for model evaluation.

Although, developers may prefer simpler models for several reasons, Domingos made a good point regarding the performance of "complex" models. As he mentions in his article, "Ten Myths About Machine Learning:" "Simpler models are preferable because they're easier to understand, remember, and reason with. Sometimes the simplest hypothesis consistent with the data is less accurate for prediction than a more complicated one."

Using these procedures, one has to bear in mind that the aim is then to not compare between models yet different algorithms that produce different models on the training folds.

Regardless of the techniques used, there should be an alignment between these models and human values. At a high-level, the main approach to alignment focuses on engineering a scalable training signal for very smart AI systems that is aligned with human intent. It has three main pillars:

- *Training AI systems using human feedback:* The aim is to train a class of models derived from pre-trained language models so that they are trained to follow human intent: both explicit intent given by an instruction as well as implicit intent such as truthfulness, fairness, and safety.
- *Training AI systems to assist human evaluation:* In order to scale alignment, engineers prefer to use techniques like recursive reward modeling (RRM), debate, and iterated amplification.
- *Training AI systems to do alignment research:*As there is indefinitely no scalable solution a more pragmatic approach might be building and aligning a system that can make faster and better alignment research progress than human-beings can.

The ultimate goal is to train models to be so aligned that they can off-load almost all of the cognitive labor required for alignment research.

Importantly, there is a need for "narrower" AI systems that have human-level capabilities in the relevant domains to do as well as human-beings on alignment research.

There are various other techniques to update model parameters, yet given the scope of this study, the next section will explain related techniques for developing recommender systems.

## Recommender Systems

The main goal of a recommender system is to produce features from both videos and text (i.e., the user question), jointly allowing their corresponding inputs to interact.

One main challenge for recommender systems relates to the use of language models for ML as they may not be inherently grounded in the physical world due to the lack of interaction during the training process.

Another challenge is embedding videos into deep learning models such as recommender systems which require more sophisticated solutions such as objects in a scene, as well as temporal information, e.g., how things move and interact, both of which must be taken in the context of a natural-language question that holds specific intent.

To improve the quality, one can collect ground-truth labels from the platform dataset and categorize factors that affect quality perception into three high-level categories: (1) content, (2) distortions, and (3) compression.

Moreover, security and privacy threats of ML models should also be taken into account as adversaries can stage effective attacks against these systems and potentially obtain sensitive information used in training the models.

During training, ML models go through episodes, each of which consists of a trajectory or sequence of actions and states.

A more effective way for automatically designing deep learning architectures could be to define a *search space*, made up of various potential building blocks that could be part of the final model. In this way, the *search algorithm* finds the best candidate architecture from the *search space* that optimizes the *model objectives*, e.g., classification accuracy.

Within the light of this information, amodel architecture can be developed in two stages:

- **Search:** In the search stage, to find an optimal path for each domain jointly, an individual reinforcement learning (RL) controller is created for each domain, which samples an end-to-end path (from input layer to output layer).
  At the end of the search stage, all the sub-networks are combined to build a heterogeneous architecture for the model.
- **Training:** For this to work, it is necessary to define a unified objective function for all the domains. An algorithm that adapts throughout the learning process should be designed such that losses are balanced across domains.

This is an efficient solution to build a heterogeneous network to address the data imbalance, domain diversity, negative transfer, domain scalability, and large search space of possible parameter sharing strategies for machine learning.

## Design Methodology

Using the co-design approach, this study explores how to understand the reliability of a model in novel scenarios.

Co-design generally refers to the collaboration between researchers and users to produce digital artefacts (Barbera et al., 2017; Cober et al., 2015). The co-design process differs from other methods of design in that it operates "bottom-up" with users being active participants in the design process, who provide critical insight into daily work practices and the existing context.

This study suggests three general categories of requirements for designing reliable machine learning (ML) systems:

(1). they should accurately report uncertainty about their predictions (*"know what they don't know"*);
(2). they should generalize robustly to new scenarios (distribution shift); and
(3). they should be able to efficiently adapt to new data (adaptation).

The process model of a co-design session is divided into three phases: establishing context, design, and presentation.

**Phase One: Establishing Context**

The objective of this stage is to create a shared understanding of aspects such as content, resources, and challenges of the recommender system to be developed.

**Phase Two: Design**

Individual team members are encouraged to make proposals, adapt, or expand on ideas and ask questions. A common trend among both high- or low-structured approaches to co-design is the use of physical artifacts to support a shared understanding of the design (Barbera et al., 2017).

**Phase Three: Presentation and Documentation**

During this process, the researchers document, both visually and in text form, the design decisions that are agreed on by the group. At the end of stage three, an initial design has been created, agreed upon, and translated into a graphical format, which can be clearly interpreted by all parties involved.

## Recommendations

When it comes to designing ML models such as recommender systems, the following suggestions can be taken into account:

**1. Defining the goal:**
Goals can include:

- user retention,
- increased revenue,
- cost reduction.

If the global task of a recommender systemis to select a shortlist of content from a large catalog one choice might be to focus on the history of the user's interaction with the service. Yet, "good recommendations" from a user perspective and from a business perspective are not always the same thing.

**2. Finding the Optimal User Touch point:**

When a decision has been made on the global goal, the best way to display recommendations need to be made. For instance, in the case of recommender systems, display of recommendations can occur:

- in the feed
- push notifications,
- email newsletter,
- the section with personalized offers in a personal account,
- or other sections on the site/application.

Many factors influence the choice of touchpoint such as push notifications or the complexity of integration with ML microservice.

As ML should ideally be implemented when it is seen by the maximum number of users, using ML at this point would be impractical.The main rule here is to integrate ML where it will make the biggest increase in business metrics.

**3. Collecting Diverse Feedback:**

Feedback is the actions a user can take to demonstrate how they feel about the content. To build a recommender system, there is a need to learn how to collect different types of feedback:

**Explicit**:
This can be a rating by any scale or a like/dislike.

I**mplicit**:
This can include;

- the amount of time a user spends on the content,
- the number of visits to the content page,
- the number of times one shares the content on social networks or sends it to friends.

Feedback should correlate with the business goals of the recommender system.

Some important technical points to consider are:

- *Expanding user feedback channels:*In addition to the time spent on the page, one can start collecting user comments and determining their tone.
- *Keeping a history of user feedback:* This helps one to identify insights in long-term users' behavior. Also, a large amount of historical data will allow one to compare models without running AB tests, in an offline format.
- Collecting data on all platforms

**4. Defining Business Metrics:**

ML experts got used to working with the metrics of ML algorithms: precision, recall, etc.businesses might be interested in other indicators such as:

- session depth,
- conversion to click//view,
- retention,
- average click per user

**5. Segmenting Users:**

From a business perspective, the audience of the site can be very heterogeneous in various ways including;

- socio-demographic characteristics,
- activity on the service (number of feedback, frequency of visits),
- geo-positions,

The system should provide the ability to calculate metrics in different user sections, to notice the improvement (or deterioration) of metrics in each particular segment.For example, there can be two large segments:

- "high activity" — users visit the app frequently and watch a lot of content,
- "low activity" — users visit the app rarely.

**6. Determining the Right Offline Metrics:**

When the feedback data is collected and the business metrics are selected, there is a choice of offline metrics, which can optimize a model, such asprecision@k or recall@k,

One can choose offline metrics that correlate with business metricsby calculating the correlation between offline and online metrics.

**7. Creating a Baseline Model:**

Rather than trying to use the most complex models to solve the problem one can start with simpler approaches instead of neural networks. This simple model is called a baseline.

For example, one can consider using a simple approach based on the k-Nearest neighborhoods algorithm to create a service for recommending content in push notifications, and only in the second iteration, move to a more complex boosting model.

**8. Choosing the ML Algorithm and Discard the Worst Models:**

The next step is to train more complex models. Recommender systems usually use both neural networks and classical ML algorithms:

- Matrix factorization,
- LogisticRegression,
- KNN (user-based, item-based),
- Boosting.

One can also prefer to count offline metrics already accumulated in the feedback system. In this way, one can distinguish very bad models from "not quite bad ones" in order totest the "not quite bad model".

**9. Run Everything Through the AB Testing System:**

Without good analytics, one either can't see the effect of a recommender system, or one can misinterpret the data, which can cause business metrics to deteriorate.

This is why AB tests need to measure both short-term and long-term effects.When conducting AB tests, one need to ensure that the samples in the test and control groups are representative.

**10. Remember the Classic Problems in Production:**

When rolling out the algorithm "in production" it is necessary to provide a solution to a number of classic problems.

- *Users' cold start:* What to recommend to those who haven't left feedback? One can make lists of globally popular content and make them as diverse as possible to be more likely to "hook" the user.
- *A feedback loop:* One can show the content to the user, then collect feedback, and run the next learning cycle on that data. In this case, the system learns from the data it generates itself. To avoid this trap, one can usually allocate a small percentage of users who receive random output instead of

recommendations — with this design, the system will be trained not only on its own data but also on users' interactions with randomly selected content.

Each of these suggestions might have their own advantages and dis-advantages. Based on the context and requirements of the model to be developed, relevant methods can be selected. It should be taken into account that there is no single one, best approach that would work for all models.

## Conclusion

Design is a complex process and designing for ML models can certainly be much more complex. Despite the progress made in the field and the rise of large-scale pre-training models, the data confronting models in real-world settings rarely match the training distribution which can ultimately affect the reliability of a model.

This study explored the use of co-design approaches for developing reliable design frameworks for deep learning in the context of recommender systems. It aims to raise awareness on how to develop reliable ML models.

While much work needs to be done in this field, the study provides some practical steps to be followed when it comes to developing reliable models. Although the study provided an architecture and model recommendations for a recommender system, the main points should be relevant for any other ML-driven model as well.

## References

[1]. Assuncao, Marcos D., et al. "Big Data Computing and Clouds: Challenges, Solutions, and Future Directions." Journal of Parallel and Distributed Computing, vol. 79–80, Dec. 2013, https://doi.org/10.1016/j.jpdc.2014.08.003.

[2]. Athanases, Steven Z., et al. "Fostering Data Literacy through Preservice Teacher Inquiry in English Language Arts." The Teacher Educator, vol. 48, no. 1, 2013, pp. 8–28.

[3]. Bargagliotti, Anna, et al. Pre-K-12 Guidelines for Assessment and Instruction in Statistics Education II (GAISE II): A Framework for Statistics and Data Science Education. American Statistical Association, 2020, https://www.amstat.org/docs/default-source/amstat-documents/gaiseiiprek-12_full.pdf.

[4]. Bersin Insights Team. Insights from IMPACT 2018. Deloitte Development LLC, 2018, https://www2.deloitte.com/content/dam/Deloitte/ca/Documents/audit/ca-audit-abm-scotia-insights-fromimpact-2018.pdf.

[5]. Bhargava, Rahul, et al. "Beyond Data Literacy: Reinventing Community Engagement and Empowerment in the Age of Data." Data-Pop Alliance, Data-Pop Alliance, Nov. 2015, https://datapopalliance.org/item/beyonddata-literacy-reinventing-community-engagement-and-empowerment-in-the-age-of-data.

[6]. Cowell, Matt. "A Roadmap for Creating a Data Literacy Program - QuantHub." QuantHub, QuantHub, 18 June 2020, https://quanthub.com/data-literacy-program. DataLiteracy. "The Data Literacy Score | Data Literacy."

[7]. Data Literacy | Learn the Language of Data, Feb. 2021, https://dataliteracy.com/data-literacy-score. ---. "About This Project - DataLiteracy.Ca." DataLiteracy.Ca, Intyernet Archive, 22 Dec. 2021, https://web.archive.org/web/20211222131031/http:/dataliteracy.ca/about-this-data-literacy-project.

[8]. Deahl, Erica. Better the Data You Know: Developing Youth Data Literacy in Schools and Informal Learning Environments. Massachusetts Institute of Technology, 2007, https://cmsw.mit.edu/wp/wpcontent/uploads/2016/06/233823808-Erica-Deahl-Better-the-Data-You-Know-Developing-Youth-DataLiteracy-in-Schools-and-Informal-Learning-Environments.pdf.

[9]. Duncan, Alan D., Donna Medeiros, Aron Clarke, et al. "How to Measure the Value of Data Literacy." Gartner, Gartner, Apr. 2022, https://www.gartner.com/en/documents/4003941. ---. "Toolkit: Data Literacy Individual Assessment." Gartner, Gartner, Apr. 2020, https://www.gartner.com/en/documents/3983897.

[10]. Ebbeler, Johanna, et al. "Effects of a Data Use Intervention on Educators' Use of Knowledge and Skills." Studies in Educational Evaluation, vol. 48, Mar. 2016, pp. 19–31, https://doi.org/10.1016/j.stueduc.2015.11.002.

[11]. Kleitz, Lauren, and Joe Shelley. EDUCAUSE Data Literacy Institute. EDUCAUSE, 24 Jan. 2022, https://events.educause.edu/educause-institute/data-literacy-institute/2022/online-1.

[12]. Kolbe, Richard H., and Melissa S. Burnett. "Content-Analysis Research: An Examination of Applications with Directives for Improving Research Reliability and Objectivity." Journal of Consumer Research, vol. 18, no. 2, Sept. 1991, pp. 243–50, https://doi.org/10.1086/209256.

[13]. Koltay, Tibor. "Data Governance, Data Literacy and the Management of Data Quality." IFLA Journal, vol. 42, no. 4, Nov. 2016, pp. 303–12, https://doi.org/10.1177/0340035216672238.
[14]. Kumar, Vivekanandan, and David Boulanger. "Explainable Automated Essay Scoring: Deep Learning Really Has Pedagogical Value." Frontiers in Education, Oct. 2020, https://doi.org/10.3389/feduc.2020.572367.
[15]. Linn, Robert L., and David M. Miller. Measurement and Assessment in Teaching, 9th Edition. Pearson, 2005, https://www.pearson.com/us/higher-education/product/Linn-Measurement-and-Assessment-in-Teaching9th-Edition/9780131137721.html.
[16]. Lorente, Clara Llebot. LibGuides: Research Data Services: Data Types & File Formats. Oregon State University, June 2021, https://guides.library.oregonstate.edu/research-data-services/data-managementtypes-formats.
[17]. Means, Barbara, et al. Teachers' Ability to Use Data to Inform Instruction: Challenges and Supports. US Department of Education, Office of Planning, Evaluation and Policy Development, Feb. 2011, https://eric.ed.gov/?id=ED516494.
[18]. National Defence. The Department of National Defence and Canadian Armed Forces Data Strategy - Canada.Ca. Government of Canada, 3 Dec. 2019, https://www.canada.ca/en/department-nationaldefence/corporate/reports-publications/data-strategy.html.
[19]. Nikou, Melpomeni, and Panagiotis Tziachris. "Prediction and Uncertainty Capabilities of Quantile Regression Forests in Estimating Spatial Distribution of Soil Organic Matter." ISPRS International Journal of GeoInformation, vol. 11, no. 2, Feb. 2022, p. 130, https://doi.org/10.3390/ijgi1102013
[20]. Salleh, KahirolMohd, et al. "Competency of Adult Learners in Learning: Application of the Iceberg Competency Model." Procedia - Social and Behavioral Sciences, vol. 204, Aug. 2015, pp. 326–34, https://doi.org/10.1016/j.sbspro.2015.08.160.
[21]. Samuel, A. L. "Some Studies in Machine Learning Using the Game of Checkers." IBM Journal of Research and Development, vol. 3, no. 3, July 1959, pp. 210–29, https://doi.org/10.1147/rd.33.0210.
[22]. Schield, Milo. "Information Literacy, Statistical Literacy and Data Literacy." IASSIST Quarterly, 2004, http://www.statlit.org/pdf/2004-Schield-IASSIST.pdf.
[23]. Walsh, John. Implementing DND/CAF Data Strategy. Government of Canada, 26 Sept. 2021, https://publicsectornetwork.co/wp-content/uploads/2021/09/John-Walsh-PDF.pdf.
[24]. Wells, Dave. Building a Data Literacy Program. 4 Jan. 2021, https://www.eckerson.com/articles/the-dataliteracy-imperative-part-i-building-a-data-literacy-program.
[25]. Williams, Paula G., et al. "On the Validity of Self-Report Assessment of Cognitive Abilities: Attentional Control Scale Associations With Cognitive Performance, Emotional Adjustment, and Personality." Psychological Assessment, vol. 29, no. 5, 2017, pp. 519–30.
[26]. Wolff, Annika, et al. "Creating an Understanding of Data Literacy for a Data-Driven Society." Journal of Community Informatics, vol. 12, no. 3, Aug. 2016, https://doi.org/10.15353/joci.v12i3.3275.
[27]. Yi Min Lim, Delia, et al. "Datastorming: Crafting Data into Design Materials for Design Students' Creative Data Literacy." C&C '21: Creativity and Cognition, Association for Computing Machinery, 2021, pp. 1–9, https://doi.org/10.1145/3450741.3465246