1. **Design challenges**

Architecture

1. Jin et al. observe that micro-service architectures are particularly challenging when used for developing ML systems with respect to failure recovery as the failure of a service may be propagated to other services if the fault is not promptly amended ([Jin et al., 2020](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b12)). **Are you putting in place a failure recovery strategy to avoid propagation of failures?**
2. **This is particularly relevant on microservice architectures where failures can propagate from one service to other services (**[**Jin et al., 2020**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b12)**).**
3. In his paper, Serban observed that heterogeneous redundancy is hard to achieve in ML-based systems ([Serban, 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b23)). This is because only a few algorithms achieve the needed accuracy for ML tasks and all these algorithms exhibit the same weaknesses ([Serban, 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b23)). **Is heterogeneous redundancy important in your system? It can be challenging since only few algorithms achieve the needed accuracy for ML tasks and they share the same weaknesses (**[**Serban, 2019**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b23)**).**
4. Kusmenko et al. argue that the use of [neural networks](https://www.sciencedirect.com/topics/computer-science/neural-network) with ML as reusable [building blocks](https://www.sciencedirect.com/topics/computer-science/building-blocks) with clear interfaces is still challenging ([Kusmenko et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b13)). **Is modularity an important quality attribute? It can be difficult to reuse ML building blocks with clear interfaces ([Kusmenko et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b13)).**
5. Muzaffar et al. investigate the use of ML for building mobile robots and conclude that the selection and design of an architecture for a mobile robot that satisfies both functional and [quality attribute requirements](https://www.sciencedirect.com/topics/computer-science/quality-attribute-requirement) is a big challenge due to less available knowledge of computer algorithms and probability theory ([Muzaffar et al., 2015](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b18)). ML knowledge is important for developers that need to understand the data and the ML model, interviewee I4 says. Bhat et al. argued that it is challenging to identify and quantify architectural expertise in ML systems ([Bhat et al., 2018](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b6)). **Do you have in the project the needed knowledge on ML?** **ML knowledge is important for developers that need to understand the data and the ML model (Bucaioni et al., 2023).** **Be aware that specific domains can require specific knowledge, like mobile robotics can require knowledge in computer algorithms or probability theory to select and design an architecture that satisfies both functional and** [**quality attribute requirements**](https://www.sciencedirect.com/topics/computer-science/quality-attribute-requirement) **(**[**Muzaffar et al., 2015**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b18)**).**
6. Serban et al. noticed that the use of ML components may introduce uncertainty when evaluating the reliability of [software architecture design](https://www.sciencedirect.com/topics/computer-science/software-architecture-design) ([Serban et al., 2020a](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b24)). Prior information on the uncertainty of ML components employed at design time is often incomplete and their usage can influence other components in the system. ML design smells are challenging to find ([Sculley et al., 2015a](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b22)). Design smells manifest in several ways. For example, using multiple languages in the development of ML systems often increases the cost of effective testing and makes it more difficult to transfer ownership to other team members ([Sculley et al., 2015a](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b22)). Another design smell is that maintaining the prototyping environment is costly, and small scale rarely reflects reality at full scale ([Sculley et al., 2015a](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b22)). One interviewee (I7) identified run-time architectures for ML as challenging with respect to the distributed execution of the ML model. **Is reliability an important quality attribute? The use of ML components may introduce uncertainty when evaluating the reliability of** [**software architecture design**](https://www.sciencedirect.com/topics/computer-science/software-architecture-design) **(**[**Serban et al., 2020a**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b24)**).   
   Do you have design smell problems? Design smells manifest in various ways, but identifying them is difficult (**[**Sculley et al., 2015a**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b22)**). Examples of design smells are: (i) using multiple languages in the development of ML systems often increases the cost of effective testing (**[**Sculley et al., 2015a**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b22)**), (ii) maintaining the prototyping environment is costly (**[**Sculley et al., 2015a**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b22)**), (iii) small scale rarely reflects reality at full scale (**[**Sculley et al., 2015a**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b22)**), and (iv) identifying the runtime architecture of the distributed execution of the ML model (Bucaioni et al., 2023).**

Data

1. Castellanos et al. identified data management, e.g., pre-processing and preparation, as significant design issues in ML systems with consequences lasting even after the system development. For example, to infer contextual information on extensive data as scientific imaging is challenging and time-consuming as a large amount of similar data is needed as input to the ML training system. **Do you have a large-enough dataset for training the ML component? (Castellanos et al., 2020) identified data management aspects as significant design issues in ML systems with consequences lasting even after the system development. For example, to infer contextual information on extensive data as scientific imaging is challenging and time-consuming as a large amount of similar data is needed as input to the ML training system.**
2. *He indicated that the cleaning of the data is cumbersome, but crucial for guaranteeing that it is not biassed and it does not break the algorithm.* ***Is your dataset clean and free from potential bias? A clean dataset is crucial for guaranteeing that it is not biassed and it does not break the algorithm.***
3. He stated that data visualisation is also challenging as bioinformatic information is not common, and current techniques are rarely applied. **Do you have applicable data visualisation techniques? Current techniques may not be applicable with specific information such bioninformatic.**
4. I8 argued that data preparation is difficult especially making statistics of it. **Do you have applicable data preparation techniques? Making statistics can be difficult without data preparation techniques.**
5. Sculley et al. looked at the dependencies between lines of code and configuration and concluded that in an ML system that is actively being developed, the number of configuration lines tends to considerably outnumber the number of lines of code and such an increment in configuration lines is challenging to handle. **Did you configure the system properly? (Sculley et al., 2015a) observed that the number of configuration lines tends to considerably outnumber the number of lines of code and such an increment in configuration lines is challenging to handle.**
6. Data observability was reported as challenging by I1 when dealing with large ML models. **Do you have applicable data observability techniques?**
7. One interviewee (I3) noted that one of the main challenges related to data is how to ensure privacy and confidentiality. **Do you have applicable techniques for ensuring privacy and confidentiality of your data?**
8. Data accuracy and completion of data are crucial when training the models of ML systems. **Is your data accurate and complete?**
9. Cloud is regarded as not suitable when a high volume of data is handled. **Do you have a proper infrastructure with respect the size of your data set? Cloud is regarded as not suitable for a high volume of data.**
10. One interviewee (I7) stated that establishing a proper infrastructure for, e.g., storing, accessing, and updating data efficiently, is also one of the main challenges. **Do you have proper infrastructure in place?**
11. Biondi et al. remarked that ML systems have high demands on setting up and maintaining the prototyping environment as small-scale environments usually do not reflect reality in full scale. **Do you have a suitable prototyping environment? (Biondi et al., 2019) remarked that ML systems have high demands on setting up and maintaining the prototyping environment as small-scale environments usually do not reflect reality in full scale.**
12. Related to this, another interviewee acknowledged as challenging of the use of batching scoring for saving in the database.

Evolution

1. Möstl et al. recognised that cyber–physical systems (CPSs) developed using ML have design challenges in managing the system and its [operational environment](https://www.sciencedirect.com/topics/computer-science/operational-environment) needs for continuous change and evolution ([Möstl et al., 2018](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b16)). As an example, they mention the [interaction design](https://www.sciencedirect.com/topics/computer-science/interaction-design) challenges related to the dynamic allocation of software when multi-core architectures are used ([Möstl et al., 2018](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b16)). **Do you have a proper process and strategy to manage the system and its** [**operational environment**](https://www.sciencedirect.com/topics/computer-science/operational-environment) **needs for continuous change and evolution? It can be challenging especially for cyber–physical systems (CPSs) developed using ML ([Möstl et al., 2018](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b16)).**
2. [Baylor et al. (2017)](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b4) and an interviewee (I9) refer to updatability as one of the main challenges. In fact, input data changing over time would require updating ML-based systems since updates in models can impact system outcomes and its performance ([Baylor et al., 2017](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b4)). **Are you engineering your ML-based system so as to adapt to input data changes, also known as concept drift?**
3. The scalability of the model was reported as a design challenge by one interviewee (I12). In particular, the interviewee remarked that when ML models do not scale, the engine can get busy and lose some requests. **Are you considering the possibility of scaling the model over time? When ML models do not scale, the engine can get busy and lose some requests (Bucaioni et al., 2023).**

QA

1. Formal verification is either impossible or impractical. **If you need formal verification, do you have proper formal verification techniques in place? Formal verification is either impossible or impractical.**
2. ML systems that are sensitive to variations in distribution.
3. limited scenario testing.
4. fault tolerance
5. One interviewee (I1) remarked that being able to explain why an ML system makes a prediction is an important challenge especially when the size of the model is increasing. **Is your ML system explainable? Explain why an ML system makes a prediction is an important challenge especially when the size of the model is increasing.**
6. Scheerer et al. argue that deductive verification and model checking would be very useful for increasing qualities such as safety, robustness, or dependability. However, because ML is fundamentally probabilistic and non-linear in nature, methods to ensure system correctness are barely applicable and only marginally relevant ([Scheerer et al., 2020](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b20)). **If you need to ensure safety, robustness, or dependability, do you have proper techniques in place? (**[**Scheerer et al., 2020**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b20)**) noted that because ML is fundamentally probabilistic and non-linear in nature, methods to ensure system correctness are barely applicable and only marginally relevant.**
7. One interviewee (I10) stated that a lack of documentation can affect the quality of ML systems together with a lack of QA guidelines. **Do you have proper documentation or a plan to document your ML system? A lack of documentation can affect the quality of ML systems together with a lack of QA guidelines.**

Model

1. Amershi et al. identify two main design challenges related to the ML model, which are managing and versioning, and reuse ([Amershi et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b2)). First, maintaining and versioning models required for ML systems is far more complicated and demanding than maintaining and versioning other types of data in software engineering ([Amershi et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b2)). In fact, ML/AI models are more difficult to manage as separate modules than other software components because models can get entangled in complicated ways and exhibit non-monotonic error behaviours ([Amershi et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b2)). The authors claim that this issue is worsened by the lack of knowledge of the problems and best practices for ML model maintenance, due to ever-evolving research in both ML infrastructure and algorithms ([Amershi et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b2)). Schelter et al. claim that wrong management of the ML model can lead to poor performance of ML systems ([Schelter et al., 2018](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b21)). Model management involves training, maintenance, deployment, monitoring, organisation, and documentation of ML models. Incorrect model management can result in poor performances and high maintenance costs ([Schelter et al., 2018](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b21)). In addition, Serban identified other challenges related to the increasing complexity of models and their opacity ([Serban, 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b23)). **Do you have a strategy and plan to manage (e.g., training, maintenance, deployment, monitoring, organisation, and documentation), version and reuse ML models? Be aware that maintaining and versioning models required for ML systems is far more complicated and demanding than maintaining and versioning other types of data in software engineering ([Amershi et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b2)). Wrong management of the ML model can lead to poor performance of ML systems and high maintenance costs ([Schelter et al., 2018](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b21)).**
2. Secondly, model customisation and reuse may be difficult considering that some of the required abilities are not commonly found in software teams. **Do you have a strategy and plan to customise and reuse ML models? Be aware that some of the required abilities are not commonly found in software teams ([Amershi et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b2)).**

SDLC

1. Wan et al. highlighted that the reuse of traditional, non-ML, software processes is problematic ([Wan et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b26)). ML or non-ML processes have different practices with respect to requirements, design, testing/quality, process, and management. ML system architectures generally include data gathering, data cleansing, feature engineering, modelling, execution, and deployment ([Wan et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b26)). In contrast, non-ML system [architectural design](https://www.sciencedirect.com/topics/computer-science/architectural-design) is a more creative approach that implements different structural divisions of software components and provides behavioural descriptions ([Wan et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b26)). **Do you have a proper software process in place? (**[**Wan et al., 2019**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b26)**) highlighted that the reuse of traditional, non-ML, software processes is problematic (**[**Wan et al., 2019**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b26)**). ML or non-ML processes have different practices with respect to requirements, design, testing/quality, process, and management.**
2. This is also remarked by an interviewee, I11, who claims that the development of ML systems should not be driven by processes for non-ML systems.
3. The distributed architecture style is commonly favoured for ML systems due to the large volume of data. In addition, ML systems have less emphasis on low coupled components than non-ML software systems: even though various features of ML systems have distinct capabilities, development teams are closely linked ([Wan et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b26)).
4. One interviewee (I10) noticed that most practices like continuous integration are not followed and that increases the complexity at the time of system completion. **Can you use continue integration techniques for the development of your system? Continue integration techniques may decrease the complexity at the time of system completion.**
5. Similar to [Wan et al. (2019)](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b26), the interviewee claimed that the lack of available documentation negatively affects the reuse of processes. **DCQA7**
6. **Best practices**

Architecture

1. Panousopoulou et al. recommended to use in-memory distributed learning architectures to achieve sophisticated learning and optimisation techniques on scientific imaging data sets: they are very effective in removing distortion on ML scientific imaging databases. **When working with scientific imaging data sets, in-memory distributed learning architectures may help you in** **achieving sophisticated learning and optimisation techniques.**
2. The four- view architecture designing approach (i.e., the conceptual, the module, the execution architecture, and the code architecture views) ensures a better separation of concerns and, hence, decreases the development complexity of mobile robotic systems developed using ML (Muzaffar et al., 2015). **When developing ML-based mobile robotic systems, consider using the four- view architecture designing approach for a better separation of concerns and, hence, decreasing the development complexity (Muzaffar et al., 2015).**
3. Schleier-Smith et al. highlighted the benefit of using a three layers architecture for separating the business logic from ML components (Yokoyama, 2019). They show the benefit by focusing on troubleshooting ML systems that might have tightly coupled functions, e.g., inference engine derived from data, and business logic code from design. In this context, the three layers architecture (i) assists engineers in breaking down the failure and traces it to the business logic part or ML components, and (ii) allows engineers to rollback the inference engine independently of the business logic when the inference engine encounters some issues. **When developing troubleshooting ML systems, consider a three layers architecture for separating the business logic from ML components (Yokoyama, 2019).**
4. Several interviewees (I3, I5, I12) recommended the use of micro-service architectures for several reasons including the following: (i) microservice architectures enable the engineers to concentrate on building the business functionalities rather than writing glue code, (ii) microservice architectures are easier to maintain as compared to monolithic architectures as they tend to use smaller and independent components, and (iii) the low cohesion among these components helps in increasing the modifiability of ML systems (I10). **Are maintainability and modifiability important for your system? If so, consider using micro-service architecture.**
5. I10 also suggested to use the client–server pattern for reducing the risks of breaking pipelines. **Does your system contain complex pipelines subject to breaks? If so, consider using the client-server pattern.**
6. Serban et al. stated that it is best to explicitly model the intrinsic uncertainty of ML components and assess how it propagates and impacts other elements in the system at the designing stage (Serban et al., 2020a). **Can you explicitly explicitly model the intrinsic uncertainty of ML components and assess how it propagates and impacts other elements in the system at the designing stage (Serban et al., 2020a)?**
7. Burns et al. suggested using a single container, single- node patterns, and multiple-node patterns for improving reusability of components and distributed development (Burns and Oppenheimer, 2016). They argue that with the above patterns it is simpler to dis- tribute implementation across various teams and reuse components in new situations. These include the ability to upgrade components sepa- rately and the ability to write them in different languages. **Are reusability and distributed development important for your system? If so, consider using a single container, single- node patterns, and multiple-node patterns (Burns and Oppenheimer, 2016).**
8. The best practice to find design smells is to focus on three types of ML design smells that are Plain-Old- Data, Multiple-Language, and Prototype smells. **Have you checked your design against the Plain-Old- Data, Multiple-Language, and Prototype smells?**

Quality assurance

1. In their paper, Biondi et al. discussed several best practices for improving certifiability, safety, time predictability and security ([Biondi et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b7)): (i) Certifiability — use of strict and certified coding standards when developing safety-critical ML components, **Do you use strict and certified coding standards when developing safety-critical ML components?**
2. (ii) Safety — implement proper mechanisms for tolerating faults and failures that may occur in complex software routines; Concerning safety, Serban recommends to focus on guaranteeing that safety-critical systems developed using ML do not reach hazardous conditions ([Serban, 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b23)); to this end, they suggest to turn off components to reach a safe state quickly. **Do you have proper mechanisms for guaranteeing that safety-critical systems developed using ML do not reach hazardous conditions?   
   Do you have proper mechanisms for** **tolerating faults and failures that may occur in complex software routines?**
3. (iii) Security — explicitly designing and developing ML systems to defend vulnerable sections of the code that may be subject to cyber-attacks, **Are you explicitly designing and developing your ML system to defend vulnerable sections of the code that may be subject to cyber-attacks?**
4. and (iv) Time predictability — static, monitor, a-posteriori analysability, and non-analysability are the best practices for preliminary classifications, that in turn can provide confidence to the design process for ML systems. **Do you have proper mechanisms, like monitoring and a-posteriori analysability for time predictability?**
5. Li et al. recommend the use of [FLSim](https://www.sciencedirect.com/topics/computer-science/federated-learning) that is a reusable and extensible federated learning simulation framework ([Li et al., 2020](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b14)). FLSim commonly assists [deep learning](https://www.sciencedirect.com/topics/computer-science/deep-learning), machine learning frameworks, for example, PyTorch and Tensor-Flow. The authors suggest to use FLSim for creating simulators for the above mentioned type of frameworks. Jin et al. stated that the principal component analysis method is known to assist in reducing the dimension of data sets and extend the components interoperability ([Jin et al., 2020](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b12)). Sculley et al. identified best practices related to the reduction of debt for data testing, debt for reproducibility, and debt in process management ([Sculley et al., 2015a](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b22)). One interviewee (I9) remarked on the importance of following a test-driven development for QA and suggested the use of the Jupiter Notebook. Another interviewee (I6) argued that a standardisation of the training and testing process would improve the verifiability of the ML models. **Do you have a test-driven development strategy for QA and, overall, is your testing process standardised?  
   Do you have a strategy for the reduction of debt for data testing, debt for reproducibility, and debt in process management?  
   Do you have a strategy for reducing the dimension of data sets and extend the components interoperability?**

Software development life cycle

1. Anjos et al. advocated that proper documentation increases the efficiency, reusability, reproductivity, and shareability of ML-based systems and can assist in designing them (Anjos et al., 2017). **DCQA7** **(Anjos et al., 2017) advocated that proper documentation increases the efficiency, reusability, reproductivity, and shareability.**
2. Spalazzi et al. focused on the development of ML-based digital forensics systems. These systems are part of the forensic science that deals with recovering and investigating the information contained in digital devices (Spalazzi et al., 2022). For these systems, Spalazzi et al. suggested following a four phases development process that includes: seizure, acquisition, analysis, and reporting (Spalazzi et al., 2022). **For forensic systems, consider following the seizure, acquisition, analysis, and reporting phases (Spalazzi et al., 2022).**
3. Wan et al. (2019) provide various best practices related to the software development life cycle. First, they identify the activi- ties that are often relevant when architecting ML-based systems: data gathering, data cleansing, feature engineering, modelling, execution, and deployment (Wan et al., 2019). Even though various features of ML-based systems have distinct capabilities, development teams are closely linked; for example, the performance of data modelling is dependent on data processing. **Have you clearly identified the following activities: data gathering, data cleansing, feature engineering, modelling, execution, and deployment? (Wan et al., 2019)**
4. Moreover, the large volume of data often favours the selection of distributed architectures. **If you have a large volume of data, consider using distributed architectures.**
5. Muccini et al. suggested using a divide et impera approach that is to brake down the ML-based system design into sub-concerns that can be handled with proper and tailored design decisions (Muccini and Vaidhyanathan, 2021). **Can you break down your system into sub-concerns that can be handled with proper and tailored design decisions (Muccini and Vaidhyanathan, 2021)?**
6. One interviewee (I11) suggested tailoring the SDLC to accommodate prototyping.
7. The interviewee also recommended avoiding changing technology as well as hiring new resources as it may negatively affect the ML systems under development. Similarly, it can lead to changes or updates in the data architecture. **Hiring new resources or changing technology can negatively affect the development and the architecture of the ML system.**
8. Another interviewee (I12) recommends to investigate whether or not the system needs real-time capabilities before starting its design. **Does your system require real-time capabilities?**
9. When dealing with general purpose ML-based systems, I9 recommends the use of test-driven development coupled with Object-Oriented programming and Jupiter Notebook. **When developing general purpose ML-based systems, consider using a test-driven development coupled with Object-Oriented programming and Jupiter Notebook.**
10. I7 remarked the importance of a proper ML infrastructure and of the training and deployment processes. For example, Federated Learning requires a distributed architecture hence the team needs to design a distributed and dynamic system. **Have you thought on ML infrastructure and deployment processes?**

Hardware and platform

1. Baylor et al. suggested the use of TensorFlow-based learner implementation with support for continuous training and serving with production-level dependability ([Baylor et al., 2017](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b4)). **Is continuous training and production-level dependability important in your system? TensorFlow-based learner implementation can be a good solution (**[**Baylor et al., 2017**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b4)**).**
2. Similarly, Abadi et al. advocated the use of TensorFlow as a way of providing improved data visualisation ([Abadi et al., 2016](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b1)). In fact, thanks to its graphical approach, TensorFlow assists the development team in debugging the nodes reducing the need for code inspections ([Abadi et al., 2016](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b1)). **Is data visualisation an important aspect in your system? TensorFlow supports data visualisation since it assists the development team in debugging the nodes reducing the need for code inspections (**[**Abadi et al., 2016**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b1)**).**
3. One interviewee suggested the use of TensorFlow and pipelining to enhance the security of ML systems (I8) while another interviewee (I9) suggested the use of Jupiter notebook when dealing with general purpose ML-based systems. **Added comment in security.**
4. Fomin et al. recommended using ML cloud technologies for increasing the quality of cutting tool states recognition in the industry ([Fomin and Derevianchenko, 2020](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b11)).

Model

1. When building an [artificial neural network](https://www.sciencedirect.com/topics/computer-science/artificial-neural-network) in ML systems, Kusmenko et al. recommended selecting [modelling languages](https://www.sciencedirect.com/topics/computer-science/modeling-language) taking into account three specific concerns: [network architecture](https://www.sciencedirect.com/topics/computer-science/network-architecture), network training, and data set model ([Kusmenko et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b13)). Network architecture plays a pivotal role as it constrains the organisation of neurons and connections among them that define the data flow ([Kusmenko et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b13)). Once properly defined, the network architecture needs to be trained: a developer can modify the training procedure without changing the architecture or a developer can combine existing architectures and training models without changing the models at all ([Kusmenko et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b13)). Finally, the compiler needs to know where to look for training data and how to load it to train a network. Furthermore, the data set must be divided into training and test data ([Kusmenko et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b13)). **When building an** [**artificial neural network**](https://www.sciencedirect.com/topics/computer-science/artificial-neural-network) **in ML systems, how did you select** [**modelling languages**](https://www.sciencedirect.com/topics/computer-science/modeling-language)**? Three aspects should be taken into account during the selection process:** [**network architecture**](https://www.sciencedirect.com/topics/computer-science/network-architecture)**, network training, and data set model ([Kusmenko et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b13)).**
2. Amershi et al. suggested a best practice for object identification systems using ML ([Amershi et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b2)). They suggested training a partial model using existing general data sets (e.g., ImageNet for object identification) and then combining them with specialised data using [transfer learning](https://www.sciencedirect.com/topics/computer-science/transfer-learning) for better performance ([Amershi et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b2)). One interviewee (I1) suggested focusing on analysing data so as to always have a model that is correct and not obsolete. Another interviewee (I4) focused on the practice of aligning the creation and training of the model according to system requirements. **When building object identification systems using ML, are you taking care of training and performance? A best practice is to train a partial model using existing general data sets (e.g., ImageNet for object identification) and combine them with specialised data using** [**transfer learning**](https://www.sciencedirect.com/topics/computer-science/transfer-learning) **for better performance ([Amershi et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b2)).**

Evolution and Data

1. Schelter et al. proposed an evolution best practice when developing large-scale forecasting problems and time series prediction systems using ML ([Schelter et al., 2018](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b21)). They suggested training a single model per time series and retraining the models every time a new forecast needs to be created ([Schelter et al., 2018](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b21)). Some classical forecasting techniques can be used to develop similar systems, such as auto-regressive integrated moving average models, [exponential smoothing](https://www.sciencedirect.com/topics/computer-science/exponential-smoothing) methods, state-space formulation, etc. **Are you developing large-scale forecasting problems and time series prediction systems using ML? A best practice is to train a single model per time series and retraining the models every time a new forecast needs to be created ([Schelter et al., 2018](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b21)).**
2. Wnag et al. focused on ML systems for image recognition and suggested a best practice that takes into account the performance of these systems with respect to environmental changes ([Wang et al., 2010](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b27)). For example, the object recognition rate for these systems is higher when the environment is brighter, i.e., with more light. The best design practice to increase the efficiency of recognition is to adjust the algorithm parameters when the intensity of the light changes ([Wang et al., 2010](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b27)). Simultaneously, to complete image enhancement, the composition model of related algorithms may need to be reconstructed ([Wang et al., 2010](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b27)). Furthermore, all pattern [recognition algorithms](https://www.sciencedirect.com/topics/computer-science/recognition-algorithm) and the composition model builder should be capable of checking the information of related sensors when necessary ([Wang et al., 2010](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b27)). **Are you building an ML system for image recognition? A best practice is to tune the performance and accuracy of the algorithm to environmental changes, e.g., increase the efficiency of recognition is to adjust the algorithm parameters when the intensity of the light changes (**[**Wang et al., 2010**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b27)**).**
3. Grid is a data preparation technique that transforms data as other hyperparameters of the modelling pipeline. One interviewee (I2) recognised that the grid searching data preparation technique suits the processing of natural language hence the interviewee recommended best practice is to build the grids when developing ML systems for [natural language processing](https://www.sciencedirect.com/topics/computer-science/natural-language-processing). **Do you have a strategy to prepare data? Grid permits to transform data as other hyperparameters of the modelling pipeline and it is a good solution when developing ML systems for** [**natural language processing**](https://www.sciencedirect.com/topics/computer-science/natural-language-processing) **(Bucaioni et al., 2023).**
4. **Design decisions**

Architecture

1. Jin et al. proposed to use micro-service architecture as a way of decomposing a big service into discrete services that can help reduce the system coupling and provide more flexibility (Jin et al., 2020). **If you need to decompose your system into discrete services, consider using micro-service architecture. This will also reduce the system coupling and provide more flexibility (Jin et al., 2020).**
2. The interviewee (I9) advocated that micro-service architectures are a better design decision when developing ML-based natural language processing systems as they assist in performing better data cleaning that, in turn, helps the accurate parsing of the document in natural language. **For ML-based natural language processing systems, consider using micro-service architecture as they assist in performing better data cleaning.**
3. So the interviewee’s main design decision is to choose an appropriate architecture based on the input data to train. **Have you chosen the architecture considering the size of the input data to train?**
4. Muzzafar et al. suggested the use of the Siemens four views architecture when developing ML-based robot navigation component (Muzaffar et al., 2015). **For ML-based robot navigation component, have you considered using Siemens four views?**
5. The choice of an appropriate archi-tecture is highlighted as an important design decision by interviewee I7 that recommended considering the overall ML architecture as well as the run-time architecture in terms of, e.g., heterogeneous computing units. **When choosing an architecture,** **have you considered the ML architecture as well as the run-time architecture?**
6. Interviewee I10 suggested opting for a client–server architecture when developing object recognition and image processing systems as this architecture can provide low coupling and high cohesion besides enhancing the system security (for example, with the use of a firewall). **If low coupling and high cohesion are important for your system, consider use a client–server architecture. If security is important for your system, consider use a client–server architecture.**
7. Serban et al. suggested including the evaluation process of architectural safety methods before moving to the development stages when develop- ing safety-critical ML systems (Serban, 2019). They advocate that this step is crucial when existing architectural patterns are not available and new ones need to be developed (Serban, 2019). **Have you included evaluation processes of architectural safety methods?** (Serban, 2019) advocate that this step is crucial when existing architectural patterns are not available and new ones need to be developed.

Hardware and platform

1. Infrastructure as Code (IaC) is known as the method of managing and providing computer [data centres](https://www.sciencedirect.com/topics/computer-science/data-center) through machine-readable specification files, rather than actual hardware setup or interactive configuration tools. Catellanos et al. pointed out that one effective design decision when building ML systems associated with databases, servers, and other IT infrastructure is to employ IaC so as to use the same structures and rules used for code development. This can reduce cost, time and risks associated with the IT infrastructure ([Castellanos et al., 2020](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b9)). **Do you have a method to manage the entire IT infrastructure, e.g. databases, servers, etc., that is needed to build your ML system?** **A best practice is to use Infrastructure as Code (IaC) so as to use the same structures and rules used for code development (**[**Castellanos et al., 2020**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b9)**).**
2. Fomin et al. advocated that the use of cloud technologies is more efficient in cutting tools state recognition systems than other alternatives such as diagnostic feature selection using combinatorial analysis ([Fomin and Derevianchenko, 2020](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b11)). Chervyakov et al. described how to reduce the number of resources used, both in terms of hardware and time, for [Convolutional neural networks](https://www.sciencedirect.com/topics/computer-science/convolutional-neural-network) (CNN) and other ML computing operations ([Chervyakov et al., 2020](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b10)). They achieve this by proposing a CNN architecture based on Residue Number System (RNS) and a new Chinese Remainder Theorem ([Chervyakov et al., 2020](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b10)). **Do you have a strategy to reduce to reduce the number of resources used, both in terms of hardware and time?**
3. Abadi et al. recognised as beneficial the use of TensorFlow for large-scale ML systems ([Abadi et al., 2016](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b1)). TensorFlow provides a higher level of performance that is also easy to scale ([Abadi et al., 2016](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b1)). The authors recognised that other tools like [Keras](https://www.sciencedirect.com/topics/computer-science/keras)and PyTorch are better for smaller systems ([Abadi et al., 2016](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b1)). One interviewee (I12) suggested that an important design decision when architecting ML-based systems is to separate the branches for the training of the pipelines from the training of the model avoiding the so-called pipeline jungles. **Do you separate the branches for the training of the pipelines from the training of the model? This is identified as a good design decision.**

Software development life cycle

1. Their design decision is to explicitly focus on the four phases of digital forensics systems during the design-making process to achieve more de- sign simplicity than applying traditional development stages (Spalazzi et al., 2022). **When developing forensic systems, have you thought on following the seizure, acquisition, analysis, and reporting phases?(Spalazzi et al., 2022)**
2. Berquand et al. focused on a design decision for ML systems that manipulate a large amount of spatial data (Berquand et al., 2019). For such systems, they advocated that concurrent engineering methods and model-based system engineering are better design choices for analysing the space data and space mission system design as compared to, e.g., sequential engineering methods (Berquand et al., 2019). **If you are dealing with large amount of spatial data, consider concurrent engineering methods and model based system engineering.(Berquand et al., 2019)**
3. Li et al. suggested federated learning as a design decision when developing ML-based mobile computing systems (Li et al., 2020). The authors claimed that federated learning can enable continuous learning on end-user devices, reducing data loss as well as improving the preservation of data privacy (Li et al., 2020). **If you need to reduce data loss as well as improving the preservation of data privacy, use federated learning. (Li et al., 2020).**
4. Configurations of ML-based systems are hard to modify and configura- tion mistakes can be costly in terms of time, computing resources, etc. According to Sculley et al. a design decision to mitigate ML systems configuration issues is to develop all models separately (Sculley et al., 2015a). **If you have different models to develop, develop them separately to reduce ML systems configuration issues.(Sculley et al., 2015a)**
5. This will help visualise the difference in the configurations as compared to other approaches (Sculley et al., 2015a). Further, developing all models separately also helps to detect unused or redundant models in the ML-based systems (Sculley et al., 2015a). **If** **you have different models to develop, develop them separately to detect unused or redundant models in the ML-based systems. (Sculley et al., 2015a).**
6. One interviewee (I1) suggested that an important design decision is to align the design of the ML systems to the related business goals. To ensure this, the interviewee suggests the use of prototyping hence the fast realise of system prototypes that can be sued to collect feedback from stakeholders with respect to business goals fulfilment. **Consider prototyping for aligning the design of the ML systems to the related business goals**
7. I7 focused on how to enable continuous ML by focusing on the ML development architecture and run-time architecture with heterogeneous computing units (e.g., CPU, GPU).
8. I3 argued that a good decision is to leverage the practical experience of ML developers as a complement to the expertise of architects. **Consider building heterogeneous teams for maximising knowledge sharing.**

Model

1. Two interviewees agreed that one important design decision is the selection of a model. Interviewee (I6) suggested selecting a model depending on the ML systems domain type for example batch system or real-time system. Such a design decision is also suggested by Amershi et al. that add that a wrong model may not help in fulfilling the requirements ([Amershi et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b2)). For example, they claim that the EfficientNet model is better than the [ResNet](https://www.sciencedirect.com/topics/computer-science/residual-neural-network) model for [image classification](https://www.sciencedirect.com/topics/computer-science/image-classification) systems due to the scaling of image dimensions by a fixed number of layers that can provide better performance in real-time ML systems ([Amershi et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b2)). **Are you putting the needed attention in selecting the model?**
2. Kusmenko et al. advocated that a design decision to achieve better performance concerns modelling and training neural processes in ML-based systems ([Kusmenko et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b13)). They argued that it is preferable to have a domain-specific language (DSL) rather than dealing with low-level constraints. For example, deep learning technologies have been more accessible by expressing layered structures as YAML or prooftext descriptions or offering high-level Python interfaces ([Kusmenko et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b13)). Interviewee I5 argued that it is important for the development team to first make a general model and then retain it and avoid newly hired team members changing it as this would impact the overall performance. **Do you have a strategy to achieve better performance? The use of domain-specific language (DSL) rather than dealing with low-level constraints is identified as a good design decision ([Kusmenko et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b13)).**

Data

1. With respect to data design decisions, Panousopoulou et al. suggested adding data visualisation techniques in the design process as it can help express the relationships between data and computing tasks ([Panousopoulou et al., 2018](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b19)). In fact, large ML-based learning systems require a special focus on facilitating [data analytics](https://www.sciencedirect.com/topics/computer-science/data-analytics) [Panousopoulou et al. (2018)](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b19). Similarly, one interviewee (I7) stated that the main design decisions to take when building ML-based systems are on how to manage and access data. I2 suggested always building the grids for ML-based natural language systems to enhance their performance. Particularly, the interviewee suggests using already existing solutions to reduce time consumption. **Do you make use of data visualisation techniques? It can help express the relationships between data and computing tasks ([Panousopoulou et al., 2018](https://www.sciencedirect.com/science/article/pii/S0164121223002558" \l "b19))  
   Do you have a strategy on how to manage and access data?  
   Are you investigating the use of existing solutions to reduce time consumption?**

Evolution

1. Components-based ML distributed systems are multi-node ML systems that increase efficiency and improve performance by handling large-scale input data and ML components (Burns and Oppenheimer, 2016). For these systems, Burns et al. suggested the use of the multi- node approach as it allows disturbed ML systems components to be upgraded independently (Burns and Oppenheimer, 2016). **For components-based ML distributed systems, consider using multi- node approach for better upgradability.(Burns and Oppenheimer, 2016).**
2. This allows the development team to handle the component as it grows even using different ad hoc technologies or languages (Burns and Oppen- heimer, 2016). **If you need to use different technologies for different components, consider using multi- node approach for heterogeneous technologies. (Burns and Oppenheimer, 2016).**
3. Wang et al. suggested design decisions for the develop- ment of ML-based mobile robots using visual capabilities (Burns and Oppenheimer, 2016). Their main design decision is to account for parameter changes in relation to environmental scenarios (Wang et al., 2010). For example, they suggested that the noise-cleaning algorithm should update its parameters to adapt to changes in light intensity (Wang et al., 2010). **When developing ML-based mobile robots using visual capabilities, consider accounting for parameter changes in relation to environmental scenarios. For example, (Wang et al., 2010) suggested that the noise-cleaning algorithm should update its parameters to adapt to changes in light intensity (Wang et al., 2010).**
4. I4 suggested considering the evolution of the ML system accounting for input data change and updating the ML system software architecture accordingly.

Quality assurance

1. When it comes to design decisions for QA, Biondi et al. suggested the use of strict coding standards and the use of (safety) certification from authorised bodies ([Biondi et al., 2019](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b7)). **Is your system critical? It can be wise (i) the use of strict coding standards and the use of (safety) certification from authorised bodies (**[**Biondi et al., 2019**](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b7)**).**
2. Baylor et al. recognised model validation as an essential phase to predict how a learning algorithm will behave on new data ([Baylor et al., 2017](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b4)). They suggest (i) always checking the validation results before pushing the algorithm into the production environment, and (ii) combining model validation with data validation to better detect corrupted training ([Baylor et al., 2017](https://www.sciencedirect.com/science/article/pii/S0164121223002558#b4)). **Are you performing the validation of the model, e.g., to predict how a learning algorithm will behave on new data? Are you combining model validation with data validation to better detect corrupted training?**