A-SIT Technology
E-Government Innovation Center

22 February 2023 Overview · AI · E-Government · 18 min read

Secure development of AI services

Bianca Danczul



Authors	Bianca Danczul
Version	2023-02-15
Source	https://technology2.a-sit.at/en/publications/secure-development-of-ai-services/
Note	Machine translation.

Key findings

- Classic development frameworks cannot be transferred one to one to AI services, for example red teaming and auditing; explainability is a new problem.
- Each AI phase has typical attacks: training (data poisoning, bias), model (model extraction), application (evasion), inference (data leakage).
- A Responsible AI Development Lifecycle adds design review, harm modelling, penetration testing and incident management to the development cycle.
- AI has no standardised requirements for documentation and auditing yet, and AI incidents are rarely reported for fear of reputational damage.
- Surveys of ML practitioners show little knowledge of attacks and protective measures; there is a strong need for further research.

AI-based services can reduce time and costs on the part of government agencies and private companies, while increasing user satisfaction. However, by their very nature, they also carry risks, especially when personally identifiable information (PII) is processed. Accordingly, it is important that security and compliance requirements are considered and adhered to during their development.

In traditional software development, there are frameworks for this purpose that simplify the secure development of services. However, these cannot be applied one-to-one to AI-based services. The goal of this study is therefore to find out which options exist for the secure development of AI services and whether their implementation is also carried out in practice. The results are reviewed and room for further research as well as proposed solutions are identified.

1. Introduction

Artificial intelligence (AI) is about to revolutionise the world we live in and has brought significant progress in both the private and the public sector. Owing to the increasing use of AI applications and given today's connected world, in which cyber threats and data breaches are becoming ever more frequent, the importance of the secure development of AI applications cannot be overstated.

Developing secure AI applications is crucial to protect the integrity of the application, prevent data breaches and safeguard users' privacy. The potential impact of a security breach in an AI application can be severe and lead to considerable financial and reputational losses. It is therefore important to consider security requirements throughout the entire development process in order to ensure the security and reliability of AI applications and to strengthen the trust of users who rely on these applications in their everyday lives.

In conventional software development, secure application development comprises a number of measures, including the use of secure programming practices, the inclusion of reliable authentication and access control mechanisms, the use of encryption and the implementation of effective vulnerability management procedures [1]. Moreover, secure software development is not a one-off but an ongoing process that includes regular security audits and penetration tests to identify and remedy potential vulnerabilities.

However, it is unclear whether this approach is feasible and sensible for AI-based applications and whether it is implemented as such in practice. The aim of this paper is therefore to find out which options exist for the secure development of AI services and whether these are adhered to in practice.

2. Overview of Characteristics of Classical and AI-Based Software Development

Classical software development and the development of AI applications have much in common, but also show some significant differences, in particular with regard to complexity and security aspects.

2.1. Similarities

Both in classical and in AI-based software development, a general software development cycle should be followed, which usually comprises phases such as requirements gathering, design, implementation, testing and maintenance [2–4]. In addition, both variants of development require collaboration with development

teams, project managers and stakeholders in order to develop high-quality software products that meet the users' requirements. Finally, software testing is indispensable to ensure that the developed software is robust, reliable and, in the case of an AI-based application, free of bias.

2.2. Differences

AI development works with machine learning (ML) algorithms and neural networks, which are much more complex than conventional software code [2]. In addition, the approach differs from conventional software development and comprises, for example, the phases training, modelling, application and inference [5]. All of this requires a deep understanding of statistical modelling, data analysis and machine learning concepts. Moreover, AI applications are often trained on large datasets, which can lead to privacy and security concerns if the data contains sensitive information.

If attackers gain control over the training data or the model, they may be able to manipulate the AI system to their advantage [6–10].

Furthermore, AI applications often require large amounts of computing resources, including high-performance computing clusters and specialised hardware such as GPUs. This can make AI development more expensive and more resource-intensive than conventional software development. Finally, AI systems are frequently used in sensitive areas such as medical diagnoses, financial transactions or education [11], which means that errors or security vulnerabilities can have serious consequences.

3. Secure Development of AI Systems in Practice

After this rough overview, a detailed insight into the secure development of AI systems in practice is to be given. First, some security concerns are mentioned, before options for the secure and trustworthy development of AI applications are listed. Finally, the practical perspective of developers is considered.

3.1. Risks and Security Concerns

Both conventional software and AI applications can be endangered by security threats such as hacking attacks, malware and data breaches. However, because of their complexity and their data-driven nature, AI systems can raise additional security concerns.

According to Khisamova et al. [12], there are two main problems associated with the use of AI that can raise security risks and ethical concerns, namely first the collection, analysis and processing of data and second AI decisions based on general data. A particular problem here is that the confidentiality of information often cannot be guaranteed, especially when vulnerabilities are present in the software. Such vulnerabilities can either lead to accidental errors or give malicious actors the opportunity to exfiltrate sensitive data or to manipulate the training data or the model in order to introduce bias or make the system less accurate [5, 7]. In areas such as medical diagnosis or criminal justice, this can have serious consequences. AI systems can also be vulnerable to attacks in which an attacker deliberately manipulates input data to cause the system to make wrong decisions [7].

Li and Zhang [11] address possible problems in more detail and identify three potential areas: security problems, privacy problems and ethical problems. Security problems include those arising from misuse of the technology itself, those arising from technical errors and those arising from the AI itself, should the AI gain consciousness. Privacy concerns can arise in the areas of data collection, data use in the cloud or

knowledge extraction. As ethical problems, it is pointed out that robots cannot follow social rules, that the role of robots is not clearly defined and whether it is ethically justifiable to destroy a robot that has developed consciousness.

Finally, Dilmaghani et al. [5] give a precise overview of which phase of AI development is vulnerable to which attack vector. The training phase is particularly vulnerable to data leakage, bias in data and data poisoning, whereas the model itself is highly vulnerable to data leakage and model extraction attacks. The application phase is exposed to evasion attacks, while the inference phase is, in turn, particularly vulnerable to data leakage.

To counter the security concerns discussed above, developers of AI applications must carefully design and test their systems to ensure that they are robust, accurate and secure. To this end, techniques such as data augmentation, model regularisation and adversarial training can be used to improve the accuracy and resilience of the AI system [9]. Developers must also carefully consider the impact of the data they use to train the system on privacy and security and ensure that the system is protected against potential attacks.

According to Li and Zhang [11], further solutions include an increased focus on security, privacy and ethics in research, including embedding ethical rules in AI development, better transparency and explainability of AI systems and improving the security and robustness of AI systems. In addition, AI development should be regulated more strongly and a focus should be placed on improving data protection.

The next chapter therefore deals with options for the secure and trustworthy development of AI applications.

3.2. Secure and Trustworthy Development of AI Applications

According to Wickramasinghe et al. [13], the lifecycle of AI systems consists of the following phases:

- Initiation: identifying the problem statement and the types of users as well as drafting the concept proposal
- Concept development and planning: defining the scope of the problem, identifying suitable data and other resources, and a feasibility study
- Requirements analysis: analysing user needs and requirements as well as drawing up the list of functional requirements
- Technical design: creating the system design and listing deliverables
- Development and testing: performing an exploratory data analysis, building the ML models and tuning the parameters, model validation and selection, as well as further testing and validation
- Implementation: deploying the system in a real environment and training the users
- Provisioning: carrying out performance monitoring and qualitative evaluation
- Optimisation: refining the model, interpreting the model and interactive visualisations

As can be seen, the lifecycle contains no security requirements, which is why Galinkin [4] proposes a Responsible AI Development Lifecycle to improve the security of AI applications. This Responsible AI Development Lifecycle is modelled on the classical Software Development Lifecycle and consists of the phases planning and review, design review, harm modelling, penetration testing and incident management.

- Planning and review is meant to support the development of new systems and the adaptation of existing ones by taking earlier findings into account and defining the first development steps. Further steps in this phase are business continuity planning and the documentation of plans and findings.
- Design review serves to establish the overall design of the system, including purpose, algorithmic components, data sources, logging and auditing as well as pre- and post-processing.
- Harm modelling considers hazards and identifies measures that should be taken before commissioning. Based on the identified possible vulnerabilities, the affected actors and the possible impacts, a risk assessment of the ML application is then carried out.
- Penetration testing should be carried out after completion of the AI system in order to uncover possible security gaps. The penetration test should include static code scans, dynamic vulnerability scans and scripted attacks and should ideally be carried out, as far as possible automatically, with every update
- Incident management comes into effect in the event of a problem. After determining the scope and extent of the damage, the business continuity plan is applied to resolve the problem.

Although the Responsible AI Development Lifecycle gives a good overview of which measures should be carried out to ensure a secure and responsible development of AI systems, it is kept very general and does not answer the question of how the steps could be carried out concretely. Accordingly, concrete measures and existing problems, identified by Avin et al. [14], are presented in the following:

- Red team exercises can be assigned to the "penetration testing" phase and are widespread in classical software development to have the security of systems checked by independent external experts and to eliminate any trust problems. In the field of AI development, red teaming is by nature more data-driven, which is why experts in this area are lacking [14].
- Audit trails can either be required by law or be used on a voluntary basis. Enabling audits must already be considered in the "design review" phase, as they entail high documentation requirements. The problem here, however, is that although requirements for documentation and auditing are already widespread in other domains, these do not yet exist in standardised form for AI development, which increasingly complicates the use of AI applications in sensitive areas [14]. First frameworks include guidelines for documenting certain characteristics of AI models and proposals for continuous logging together with precise documentation of the training data and results used to train the models.
- Interpretability and explainability are likewise essential aspects for softening the "black box" character of AI systems and for facilitating the assurance of the security, accountability and fairness of AI systems. This point should already be considered in the "planning and review" phase in order to define whether the use of "Explainable AI" (XAI) techniques is necessary and sensible. According to Avin et al. [14], such methods should on the one hand enable the end user to understand how a

model produces an output and on the other hand remain faithful to the model in the explanation and accurately reflect the underlying behaviour. In addition to interpretability and explainability, reproducibility enables external teams to rebuild and challenge an AI system in order to confirm the developers' claims. Initiatives such as the ACM Artifact Review and Badging and the ML Reproducibility Challenge provide incentives for reproducibility in the research environment [14].

- Privacy-preserving machine learning deals with solving privacy problems that can arise in machine learning and is to be assigned to the "harm modelling" phase. Such problems include, for example, unauthorised access to data used to train the models, the violation of privacy or unauthorised access to the trained model itself [14]. Possible techniques here are [14]:

- Federated learning techniques, which enable the centralised training of a model with decentralised data without the raw data ever leaving the source device

- Differential privacy techniques, which adapt the development process so that the trained models retain meaningful population-level statistical patterns but reduce the risk of inferring information about individuals, and

- Encrypted computation, which enables data owners and model developers to train models without either side having access to the other side's information.

There are already some open-source implementation projects, such as Opacus from PyTorch, Tensorflow Privacy, FedAI, PySyft, Flower and OpenFL, but hardly any standardised software libraries, which is why awareness among developers is rather low [14].

- Bias and security deficiencies occur increasingly, as vulnerabilities and risks are often not detected before release owing to the complexity of AI systems. In classical software development, detected vulnerabilities have long been published (e.g. CVE); in AI development, there is nothing comparable to date. According to Avin et al. [14], a large part of the knowledge about AI harms comes from researchers and investigative journalists who have only limited access to the AI systems they investigate and often maintain antagonistic relationships with the developers whose harms they uncover. This area is to be assigned to the "harm modelling" phase.

- Reporting of AI incidents is important to prevent theoretical risks from turning into actual harm and can be assigned to the "incident response" phase. For fear of reputational damage, however, incidents are rarely reported, which is why incentives for sharing are required. For example, sharing could be mandated by law, or options for anonymous sharing with a trusted third party (analogous to a CERT) could be considered. The Partnership on AI is experimenting with such a platform through its AI Incident Database, which compiles information about AI incidents from both public sources and developer reports, and the Center for Security and Emerging Technologies developed a taxonomy with three categories (specification, robustness and security) based on reported incidents, with more than 100 incidents serving as examples for each category. [14]

Finally, Dilmaghani et al. [5] go further into technical detail and propose concrete countermeasures and defences against known attacks, which should be considered in every phase of the Responsible Development Lifecycle.

- Measures against data breaches are anonymisation, de-identification and privacy-enhancing technologies (PET). Often it is enough to mask sensitive information to ensure privacy and data security, for example by applying k-anonymity measures. In more sensitive areas, the use of PET techniques should be considered. One example is the OPen ALgorithms (OPAL) project, which protects data through access protocols and aggregation procedures and allows researchers to submit the algorithms that are to be trained on the data instead of handing over the data. [5]

- Bias in data can be detected with various statistical metrics, such as mean difference, residual difference or normalised information. Further options are the use of approaches such as optimised pre-processing, learning fair representations or adversarial debiasing, or of ready-made toolboxes such as Lime or FairML. [5]

- Data poisoning, i.e. poisoned data, can in principle be detected by identifying outliers and anomalies, except when the attacker knows the data distribution and thus circumvents anomaly detection [5].

- Model extraction attacks can be prevented by the PATE technique, i.e. by training several models using different partitions of training data for each model [5].

- Evasion attacks lead to misclassification through small changes, which is why a simple defence mechanism is to ensure that a minor change to the input cannot significantly alter the result [5].

In summary, when developing an AI application, the lifecycle of responsible AI development should ideally be followed and, beyond that, concrete measures should be taken to defend against known attacks on the models and data. The next chapter therefore presents the perspective of practitioners.

3.3. Developers' Perspectives

In 2020, Kumar et al. [15] conducted a study on the industry perspective of 28 organisations in the field of adversarial machine learning. For this purpose, 56 machine learning experts who are responsible for building and securing ML models in their respective organisations were interviewed. The results of this study were sobering and show that most of the surveyed companies have neither tools nor know-how for securing ML systems and focus on traditional security rather than on AI security. Specific problems in development are in particular little knowledge of possible attacks and sparse knowledge of ML-specific secure programming practices. Further problems include little knowledge of the static and dynamic analysis of ML systems, of carrying out auditing and logging, and of detecting and monitoring ML systems. In the rollout phase, problems include in particular the lack of automated tests for adversarial attacks as well as little knowledge in the area of red teaming. In the case of a concrete attack, the authors see a need for improvement in detecting and assessing ML vulnerabilities as well as in incident response, including forensic measures and recovery.

Building on this, in 2021 Boenisch et al. [16] published a study on the security and privacy awareness of a total of 83 data scientists. The questionnaire comprised 25 questions divided into the areas demographics, data and sensitivity, ML security, attacks, ML privacy and security practices, and GDPR knowledge. The results again show that the awareness of the surveyed ML practitioners of threats as well as of ML security and privacy practices is relatively low, which may be because most ML practitioners had no academic training in ML security and privacy. Specifically, there is little familiarity with protection strategies and well-known ML security and privacy libraries.

The study by Bieringer et al. [17] from 2022 on the mental models of industry practitioners with regard to adversarial machine learning, in which 15 practitioners were interviewed, also showed similar results. It was found that the distinction between ML threats and traditional threats is often not clear to practitioners and that ML-specific threats are often regarded as less relevant than traditional threats, although there is certainly practical experience with (successful) attacks on ML models. Accordingly, the recommendations include in particular raising awareness of adversarial ML attacks, integrating measures to prevent adversarial ML into workflows and creating suitable regulatory and standardised frameworks in this area.

4. Conclusion

In summary, the secure development of AI applications is of crucial importance to protect the application and the users' data from security breaches. As AI continues to evolve and appears ever more frequently in our daily lives, it is essential to prioritise security throughout the entire development process. By implementing secure programming practices, access controls and vulnerability management procedures, the development of AI applications can be made more secure and reliable. In practice, however, this proves difficult, as established techniques such as red teaming or auditing cannot be transferred one-to-one to AI development, and new problems such as those of explainability and interpretability arise. This view is confirmed by AI developers in practice, which is why there is a strong need for research in the area of secure development of AI systems.

References

[1] R. A. Khan, S. U. Khan, H. U. Khan und M. Ilyas, „Systematic Literature Review on Security Risks and its Practices in Secure Software Development", IEEE Access, Jg. 10, S. 5456–5481, 2022, doi: 10.1109/ACCESS.2022.3140181.

[2] F. A. Batarseh, R. Mohod, A. Kumar und J. Bui, „The application of artificial intelligence in software engineering" in Data Democracy, Elsevier, 2020, S. 179–232, doi: 10.1016/B978-0-12-818366-3.00010-1.

[3] M. Howard und S. Lipner, The security development lifecycle: SDL, a process for developing demonstrably more secure software. Redmond Wash.: Microsoft Press, 2006.

[4] E. Galinkin, „Towards a Responsible AI Development Lifecycle: Lessons From Information Security", 6. März 2022. [Online]. Verfügbar unter: http://arxiv.org/pdf/2203.02958v1.

[5] S. Dilmaghani, M. R. Brust, G. Danoy, N. Cassagnes, J. Pecero und P. Bouvry, „Privacy and Security of Big Data in AI Systems: A Research and Standards Perspective" in 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, S. 5737–5743, doi: 10.1109/BigData47090.2019.9006283.

[6] Duygu Sinanc Terzi, Ramazan Terzi und Seref Sagiroglu, „A survey on security and privacy issues in big data".

[7] M. Rigaki und S. Garcia, „A Survey of Privacy Attacks in Machine Learning", 15. Juli 2020. [Online]. Verfügbar unter: http://arxiv.org/pdf/2007.07646v2.

[8] Haina Ye, Xinzhou Cheng, Mingqiang Yuan, Lexi Xu, Jie Gao und Chen Cheng, „A survey of security and privacy in big data".

[9] Gayatri Sravanthi Kuntla, „Security and privacy in machine learning. A survey“.

[10] R. Bao, Z. Chen und M. S. Obaidat, „Challenges and techniques in Big data security and privacy: A review“, Security and Privacy, Jg. 1, Nr. 4, e13, 2018, doi: 10.1002/spy2.13.

[11] X. Li und T. Zhang, „An exploration on artificial intelligence application: From security, privacy and ethic perspective“ in 2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, China, 2017, S. 416–420, doi: 10.1109/ICCCBDA.2017.7951949.

[12] K. Zarina I., B. Ildar R. und S. Elina L., „Artificial Intelligence and Problems of Ensuring Cyber Security“, 2020, doi: 10.5281/zenodo.3709267.

[13] C. S. Wickramasinghe, D. L. Marino, J. Grandio und M. Manic, „Trustworthy AI Development Guidelines for Human System Interaction“ in 2020 13th International Conference on Human System Interaction (HSI): Tokyo, Japan, 06-08 June, 2020 : online proceedings, S. Muramatsu, Hg., Piscataway, NJ: IEEE, 2020, S. 130– 136, doi: 10.1109/HSI49210.2020.9142644.

[14] S. Avin et al., „Filling gaps in trustworthy development of AI“ (eng), Science (New York, N.Y.), Jg. 374, Nr. 6573, S. 1327–1329, 2021, doi: 10.1126/science.abi7176.

[15] R. S. S. Kumar et al., „Adversarial Machine Learning – Industry Perspectives“, 2020.

[16] F. Boenisch, V. Battis, N. Buchmann und M. Poikela, „“I Never Thought About Securing My Machine Learning Systems”: A Study of Security and Privacy Awareness of Machine Learning Practitioners“ in MuC ‘21: Mensch und Computer 2021, Ingolstadt Germany, 2021, S. 520–546, doi: 10.1145/3473856.3473869.

[17] L. Bieringer, K. Grosse, M. Backes, B. Biggio und K. Krombholz, „Industrial practitioners’ mental models of adversarial machine learning“, Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022), S. 97– 116, 2022. [Online]. Verfügbar unter: https://www.usenix.org/conference/soups2022/presentation/bieringer