Machine Learning Models in Industrial Blockchain, Attacks and Contribution

Fatemeh Ghovanlooy Ghajar[1], Axel Sikora[1], Jan Stodt[2], Christoph Reich[2]

[1] Institute of Reliable Embedded Systems and Communication Electronics (ivESK), Offenburg University
{Fatemeh.Ghovanlooy,Axel.Sikora}@hs-offenburg.de
[2] Institute for Data Science, Cloud Computing and IT Security – Furtwangen University
{Jan.Stodt,Christoph.Reich}@hs-furtwangen.de

Abstract. The importance of machine learning has been increasing dramatically for years. From assistance systems to production optimisation to support the health sector, almost every area of daily life and industry comes into contact with machine learning. Besides all the benefits that ML brings, the lack of transparency and the difficulty in creating traceability pose major risks. While there are solutions that make the training of machine learning models more transparent, traceability is still a major challenge. Ensuring the identity of a model is another challenge. Unnoticed modification of a model is also a danger when using ML. One solution is to create an ML birth certificate and an ML family tree secured by blockchain technology. Important information about training and changes to the model through retraining can be stored in a blockchain and accessed by any user to create more security and traceability about an ML model.

Keywords: Machine learning; Blockchain; Traceability; Security

1 Introduction

Machine learning has successfully found its way into almost every area of everyday life. ML makes our everyday lives easier with the help of assistance systems, makes production processes more efficient, detects errors before they occur and helps to develop medicines faster than ever possible by humans. However, it must also be said that at the beginning of the increased use of ML, many people were dazzled by its capabilities and refrained from using documentation methods that were already established in other areas. And even if documentation exists, it usually has gaps that make it difficult or impossible to follow up at a later stage.

For a number of reasons, organizations wanted to stay ahead of the curve in consistently protecting ML assets. In recent years, huge firms are investing directly in machine learning for the first time. Second, standard-setting organizations such as ISO are proposing certification guidelines to assess the security of machine learning (ML) systems [1], whose recommendations have historically been sought by the industry. The European Union has even developed a detailed checklist to assess the dependability of ML systems [2]. Lastly, machine learning is rapidly becoming a company's primary value proposition.

The lack of traceability not only makes verification by external auditors difficult, but also internal auditing. Even if there is documentation of the process, it can be very difficult to find a source of error hidden in poorly documented preprocessing of the training data. One problem that is noticeable when looking at the documentation of preprocessing is that it usually only talks about using "a subset of dataset X". Which subset it is, is usually not documented. One technology that is particularly suitable for traceability is the blockchain. Information that is recorded in the blockchain can no longer be changed, and updates to the information can be traced. The decentralized storage of data also ensures that the stored data is still available even if a local copy of the data is lost. To address the traceability problems that exist in machine learning in the area of documentation, this paper introduces two concepts: the ML birth certificates and the ML family tree. Both concepts aim to clarify the ancestry of the models and to document decisions in the area of the model's origin clearly and semi-automatically.

Of course, the security and privacy problems for data (transactions) exchange and transmission in such a novel network environment are taken into account in the architecture of blockchain [3]. Several recent studies have indicated that (1) utilizing AI to improve the performance of a blockchain system [4–7] and (2) using blockchain to improve the security and privacy of data and model transfer on an AI system, [8–10] are two attractive research areas.

The study is structured as follows: part two describes the current state of the art, and section three discusses the security challenges associated with machine learning models. In the fourth part, we provide a strategy for maintaining the model's security, and in the last section, we reach a conclusion.

2 Related Work

McGraw et al. [11] identified 78 risks in ML security and highlighted the top 10 of them. Most of these attacks target the data used to train the model. Attacks also exist when the model is retrained: online learning, in which the model is trained while in use with new data, and attacks that occur during transfer learning, in which an existing model is trained (tuned) on new data. A detailed taxonomy of machine learning attacks was described by Pitropakis et al. [12]. In addition to the theory-focused reviews, Cheatham et al. [13] provide a detailed overview of the resulting consequences in the areas of individuals, organisations and society.

In this industrial use scenario, in an adversarial ML setting, transparency may need to testify across three modalities: that the ML platform is implemented securely, that the MLaaS fulfills basic security requirements, and that the ML model embedded in an edge device meets basic security objectives. Providing test harnesses to increase the security assurance of products developed on top of formal verification, such as [14] to address large-scale ML models used in industry, is an intriguing approach.

In order to capture some information about the creation of the ML model and its creation, Mitchell et al. [15] introduced the concept of ML Model Report Cards. However, it must be clearly stated that the aim of the model report cards is to create transparency in the area of fairness, not traceability. Although the model report cards (may) contain information about the creation, the level of detail to be achieved is not specified. In addition, evolution of models via online learning and especially in transfer learning, it is difficult to determine the ”base model” used in the process. Over several generations of transfer learning, the problem becomes even more acute; in retrospect, traceability is factually no longer given.
To address some of the problems mentioned, Arnold et al. [16] developed FactSheets 360, which collects more information in the area of traceability and versioning of the ML model. But even here, the quality/granularity of the collected information is not defined in detail. In examples of FactSheets 360, such vague statements as ”The test data consists of a subset of data set X” [17] are made. This coarse granularity does not allow outsiders or even internal staff to check the creation of the model afterwards. In addition, there is no automatic fact collection in the area of data pre-processing. In addition to the approaches mentioned, there are now best practices and tools in the field of ML DevOps [18] that attempt to solve the traceability problem mentioned above. For example, there are model registries that record the processing steps that have been carried out [19]. Examples of tools used for this are MLFlow[1] and Comet[2]. One problem with these tools, however, is the centralised storage approach of these solutions. This creates a large number of data silos, which makes it difficult to keep track of the data and leads to a lack of traceability when the centralised storage approach is switched off.

3 Issues

Issues of ML models exist in the areas of security against attacks, the insufficient documentation of the model creation, and tracing the lineage of ML models. In this section that follows covers the security challenges that arise throughout the construction of an ML system, as well as while the system is under assault and being readied for deployment.

3.1 Security Attacks

Attacks on ML models can sometimes have dramatic consequences [20]. Attacks on process optimization models can be used to cause financial damage by slowing down or stopping production. However, not every attack is only financially damaging. It is easy to imagine that human damage can be inflicted when decisions are made about people; making attacks on machine learning models only more critical. The following are the most significant assaults that may occur against machine learning models and software.

1. False data training model: a training model based on datasets with erroneous data. In the ML development phase, vulnerability might occur. The majority of ML developers are unaware that attackers may breach the repository of ML training datasets, poisoning the dataset [21].
2. Malicious code: typically, programmers don’t start from scratch when creating a model; instead, they utilize existing code. If an attacker uploads modified scripts to websites like library or sample code, he or she can simply manipulate the model and result. Vulnerabilities might occur at this level due to the absence of automated tools for secure developers as well as a transparency center for machine learning systems.

[1] https://mlflow.org/
[2] https://www.comet.com/site/

3. If an attacker constructs a model and utilizes it as spy software or to influence model output, they are acting as model programmers. The danger of an attacker posing as a machine learning (ML) developer exists for apps that small businesses without detection and monitoring security measures seek to utilize for minimal cost. This kind of machine learning software may be built to spy on and exploit client data.
4. Model poisoning: An attacker degrades the model performance in order to obtain a different model decision [22].
5. Manipulation of a running program and modification of its output, when the machine learning system is under assault, and the objective of the attack is something that is important to the business of the industry.

3.2 Insufficient Documentation of Model Creation

As mentioned in related work, there are approaches to improve the documentation of ML model creation and use. However, both of the approaches presented, model report cards and FactSheets 360, still have the problem that the documentation of individual model quality-determining steps (e.g., data preprocessing) are not automatically documented. Lack of information on how the data set is pre-processed makes it difficult or even impossible to check the processes at a later stage. However, this is very important in order to investigate errors in the model and to find the origin of the error. This could be due to faulty preprocessing or deliberate attack on the data.

3.3 Tracing Lineage of ML Model

Lack of information about how the data set was pre-processed makes it difficult or even impossible to check the processes later. Although there are approaches to versioning data in the area of data preprocessing, there is usually no traceability of how the data was processed, e.g. which methods, which parameters. However, this is very important to investigate errors in the model and find the origin of the error. This could be due to faulty pre-processing or a deliberate attack on the data.

4 Method

The proposed method to address the challenges comprises two parts. Create the birth certificate before adding the hash and signature to the distributed ledger. Due to the immutability of blockchain, it cannot be altered or controlled. However, it will be required to have the model on blockchain as a reference for determining the method's dependability.

4.1 First Step

To realize detailed traceability of the ML model, we introduce the idea of ML birth certificates as well as an ML family tree to trace the changes of the base ML model and the changes made. A graphical example can be seen in Fig. 1.

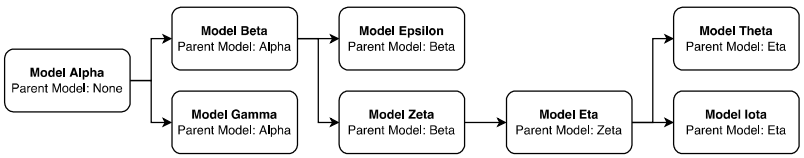

Fig. 1. ML Birth Certificates and ML Family Tree

ML birth certificates contain detailed information about the creation process, in order to make the creation process verifiable later in the life cycle of the model, if required. This information is similar to the concept of ML Model Report Cards [15] introduced by Mitchell et al. and the FactSheets 360 by Arnold et al. [16], our ML birth certificates contain a much finer granularity of information that is captured semi-automatically. For example, it logs exactly which preprocessing steps and which commands were used in which software version.
In order to make the descent of models from previous models (e.g., through transfer learning or online learning) clearly comprehensible, the concept of the ML family tree is introduced. Here, the ML birth certificate refers to the parent model. An example of the blockchain entries for the model birth certificates can be seen in Listing 1.1.

```
[
  {
    "record-type": "model",
    "model-name": "Model Alpha",
    "model-id": "8ad53dd3-4b02-4af5-a17f-e03d2fabee1c",
    "parent-model-name": "None",
    "parent-model-id": "None",
    "training-data-hash": "3cf33bd19e5001e9c151fe8127632e9...",
    "training-data-url": "ipfs://hl99h884pkragfmnlgxl",
    "payload": "exde4UOIQvV/f4bwZ59bhg=="
  },
  {
    "record-type": "model",
    "model-name": "Model Beta",
    "model-id": "402441d4-677f-4363-84aa-6f5a400a179cc",
    "parent-model-name": "Model Alpha",
    "parent-model-id": "8ad53dd3-4b02-4af5-a17f-e03d2fabee1c",
    "training-data-hash": "960f0bac5d1740c8ef0924442bc31ea8...",
    "training-data-url": "ipfs://9i0xd83ucroffgp739x4",
    "payload": "BNuEnBslKzpDk1nCwWys2A=="
  }
]
```


Listing 1.1. Example of blockchain blocks of the Model Birth Certificates

4.2 Second Step

To ensure traceability, the data used for the training is either stored as hash in the blockchain (for confidential data) or stored in IPFS [23] to ensure optimal traceability. To ensure about originality and address the concerns in Section 3.1, it is advised that the model be placed in a Docker container and stored it in IPFS. It also introduces a smart contract that verifies and signs the output based on the model's data stored in the blockchain. The process of data verification and sign by the smart contract can be seen in Fig. 2. An overview of the architecture can be seen in Fig. 3

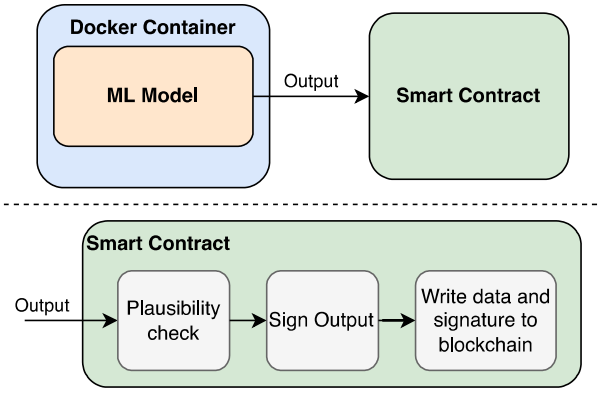

Fig. 2. Data verification and sign process of model output.

5 Conclusion

The vast majority of machine learning engineers and incident responders working in the sector do not lack the abilities essential to safeguard enterprise-grade ML systems from being attacked by malicious actors. Semi-automated data capture and blockchain technologies can help trace the lifecycle of ML models to mitigate the risks of ML model attacks and provide a more comprehensive trace of the development process. The blockchain and the concepts of ML birth certificates and ML family tree presented here also help to record the lineage of models in more detail. This means that it is always clear on which data or model basis a model was trained or retrained.

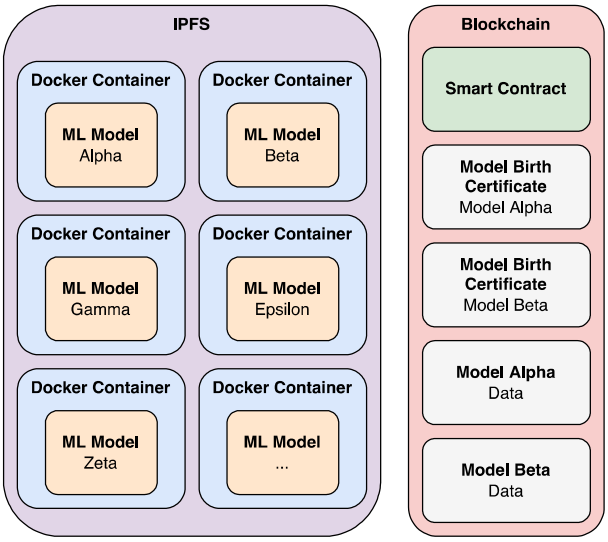

Fig. 3. Architecture.

References

1. ISO: ISO/IEC JTC 1/SC 42 - Artificial intelligence https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/committee/67/94/6794475.html.
2. Smuha, N.A.: The eu approach to ethics guidelines for trustworthy artificial intelligence. Computer Law Review International **20**(4) (2019) 97–106
3. Stodt, J., Schönle, D., Reich, C., Ghovanlooy Ghajar, F., Welte, D., Sikora, A.: Security audit of a blockchain-based industrial application platform. Algorithms **14**(4) (2021) 121
4. Marwala, T., Xing, B.: Blockchain and artificial intelligence. arXiv preprint arXiv:1802.04451 (2018)
5. Singh, S.K., Rathore, S., Park, J.H.: Blockiotintelligence: A blockchain-enabled intelligent iot architecture with artificial intelligence. Future Generation Computer Systems **110** (2020) 721–743
6. Ghovanlooy Ghajar, F., Sikora, A., Welte, D.: Schloss: Blockchain-based system architecture for secure industrial iot. Electronics **11**(10) (2022) 1629
7. Tagde, P., Tagde, S., Bhattacharya, T., Tagde, P., Chopra, H., Akter, R., Kaushik, D., Rahman, M., et al.: Blockchain and artificial intelligence technology in e-health. Environmental Science and Pollution Research **28**(38) (2021) 52810–52831
8. Lo, S.K., Liu, Y., Lu, Q., Wang, C., Xu, X., Paik, H.Y., Zhu, L.: Towards trustworthy ai: Blockchain-based architecture design for accountability and fairness of federated learning systems. IEEE Internet of Things Journal (2022)
9. Ghovanlooy Ghajar, F., Salimi Sratakhti, J., Sikora, A.: Sbtms: Scalable blockchain trust management system for vanet. Applied Sciences **11**(24) (2021) 11947
10. Lo, S.K., Liu, Y., Lu, Q., Wang, C., Xu, X., Paik, H.Y., Zhu, L.: Blockchain-based trustworthy federated learning architecture. arXiv preprint arXiv:2108.06912 (2021)
11. McGraw, G., Bonett, R., Shepardson, V., Figueroa, H.: The Top 10 Risks of Machine Learning Security. **53**(6) 57–61
12. Pitropakis, N., Panaousis, E., Giannetsos, T., Anastasiadis, E., Loukas, G.: A taxonomy and survey of attacks against machine learning. **34** 100199
13. Cheatham, B., Javanmardian, K., Samandari, H.: Confronting the risks of artificial intelligence. http://ceros.mckinsey.com/unintended-consequences-desktop.
14. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient smt solver for verifying deep neural networks. In: International conference on computer aided verification, Springer (2017) 97–117
15. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I.D., Gebru, T.: Model cards for model reporting. In: Proceedings of the conference on fairness, accountability, and transparency. (2019) 220–229
16. Arnold, M., Bellamy, R.K.E., Hind, M., Houde, S., Mehta, S., Mojsilovic, A., Nair, R., Ramamurthy, K.N., Reimer, D., Olteanu, A., Piorkowski, D., Tsay, J., Varshney, K.R.: FactSheets: Increasing Trust in AI Services through Supplier's Declarations of Conformity
17. IBM Research: AI FactSheets 360 https://aifs360.mybluemix.net/examples/aifs360.mybluemix.net/examples.
18. Rubasinghe, I., Meedeniya, D., Perera, I.: Traceability management with impact analysis in devops based software development. In: 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE (2018) 1956–1962
19. Kreuzberger, D., Kühl, N., Hirschl, S.: Machine learning operations (mlops): Overview, definition, and architecture. arXiv preprint arXiv:2205.02302 (2022)
20. Cheatham, B., Javanmardian, K., Samandari, H.: Unintended Consequences

21. Goldblum, M., Tsipras, D., Xie, C., Chen, X., Schwarzschild, A., Song, D., Madry, A., Li, B., Goldstein, T.: Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
22. Panda, A., Mahloujifar, S., Bhagoji, A.N., Chakraborty, S., Mittal, P.: Sparsefed: Mitigating model poisoning attacks in federated learning with sparsification. In: International Conference on Artificial Intelligence and Statistics, PMLR (2022) 7587–7624
23. Benet, J.: Ipfs-content addressed, versioned, p2p file system. arXiv preprint arXiv:1407.3561 (2014)