Proceedings Paper DOI: 10.58190/icat.2023.33

PROCEEDINGS OF

INTERNATIONAL CONFERENCE ON ADVANCED TECHNOLOGIES

https://proceedings.icatsconf.org/

11th International Conference on Advanced Technologies (ICAT'23), Istanbul-Turkiye, August 17-19, 2023.

Evaluation Framework for the Use of Privacy Preserving Technologies for Production Data

Lennard Sielaff[*], Ruben Hetfleisch[+], Michael Rader[+]

[*]*Fraunhofer Institute for Manufacturing Engineering and Automation IPA*
Nobelstraße 12, 70569 Stuttgart, Germany
Lennard.Sielaff@ipa.fraunhofer.de

[+]*Fraunhofer Austria Research GmbH*
Theresianumgasse 7, 1040 Wien, Austria
{Ruben.Hetfleisch, Michael.Rader}@fraunhofer.at

***Abstract* — To prevent unplanned machine downtime in production, machine conditions can be monitored and even predicted using condition and failure models based on current machine and process data. As most of these models are data-intensive, machine users often do not have enough data to develop these models themselves and want to collaborate with other companies. Since these models often require critical and classified machine and process data, which could be extracted from the models using attacks such as model inversion, sharing existing models between companies is not an option as it leaves one party vulnerable. Privacy preserving technologies such as homomorphic encryption, differential privacy, federated learning and secure multi-party computation can help overcome this problem. With the help of these approaches, there is no need to transmit sensitive data unencrypted to third parties in order to cooperate and take advantage of high-performance models. The aim of this paper is to first summarize the current state of research on privacy-preserving technologies in production, and then to provide a simple to use evaluation method and criteria. The focus is on enabling production workers to make informed decisions and exploit the full potential of existing data without the need for prior knowledge of privacy-preserving technologies. Finally, the evaluation method is validated using two example use cases in a production environment and the results are discussed.**

keywords — ***privacy preserving technologies, production data, homomorphic encryption, federated learning, decision support***

I. INTRODUCTION

Operational and process data from machines and plants is a valuable asset that can form the basis for innovative data-based services. To develop solutions that help make not only individual processes, but entire industries and value chains more efficient and sustainable, innovative collaboration across organisational units and company boundaries is required. Concerns about privacy and security often mean that this potential remains untapped [1].

One solution to this privacy and security problem is the use of privacy-preserving machine learning approaches [2], such as homomorphic encryption, federated learning, differential privacy, and secure multi-party computation. However, these new approaches are unfamiliar to most production workers faced with the challenges of using new data-based services. They are experts in their respective fields, but are often not trained or informed about these new ways of securely transferring data between organisations. Even if the employee has heard of one of the technologies, he or she is still faced with the problem of whether this approach is the right one for the problem at hand, or whether another approach might produce better results [3].

Therefore, the goal of this paper is to first summarise the current state of research on privacy-preserving technologies for production data, with a focus on inter-enterprise data transfer, and then to provide an easy-to-use method that can be used to evaluate specific use cases in order to provide decision support on whether and which of these technologies can be used. The focus of the methodology is to enable production staff with no prior knowledge of privacy technologies to make informed decisions and realise the full potential of existing data. In addition to the methodology, criteria are developed that are necessary and must be followed for the selection and design of the technology. Finally, the method is validated using two example use cases in a production environment and the results are discussed.

II. STATE OF RESEARCH

The operation of machines, which today are often equipped with a large number of sensors, generates data that, when combined with other production data, can provide valuable insights for maintenance. For example, this data can be used to identify that certain events have occurred and why they have occurred, or to predict when certain events will occur again [4]. In many cases, this data is particularly worthy of protection, for example, because it contains intellectual property [5]. In this situation, confidentiality preserving machine learning approaches are the means of choice to gain insights without

compromising the confidentiality of the data. For example, homomorphic encryption (HE) allows machine data to be analysed with third-party models without losing confidentiality. Federated Learning (FL) allows machine operators to collaborate to build an analytics model without individual machine operators having to share their data. Differential Privacy (DP) can be used to prevent confidential machine data from being extracted from published models. To compare metrics between companies without the need to publish an individual value, Secure Multi-party Computation (SMC) can be used.

A. *Homomorphic Encryption*

An encryption method is called homomorphic if there are operations on secret texts that correspond to additions and/or multiplications on the associated plaintexts. In contrast, if secret texts of non-homomorphic encryption schemes are manipulated without knowledge of the key, this will almost certainly result in invalid or randomly appearing plaintexts. In the case of "non-malleable" encryption schemes, this is even a necessary condition. Most early homomorphic encryption schemes supported either only multiplications or only additions on secret texts. Fully homomorphic encryption schemes allow arbitrary computations on secret texts [6]. This means that - at least in theory - any computation that can be performed on plaintext can also be performed on secret text. The first fully homomorphic encryption scheme introduced by [6] was highly inefficient and thus unsuitable for practical applications. Over the last few years, the field has experienced rapid development, including numerous performance optimizations [7], [8], [9], [10], [11], harnessing floating-point numbers [12], [13], or efficiently evaluating arbitrary unary functions [14]. In addition, a large number of software libraries exist today that enable the implementation of concrete use cases, e.g. SEAL, PALISADE, HElib, Concrete, TFHE, and Lattigo.

Homomorphic encryption therefore lends itself to implementing trusted inference use cases: Input data (e.g., machine data) for an machine learning model can be homomorphically encrypted and passed to a model owner, who applies his model to it and sends the homomorphically encrypted output data back to the input data owner. Thus, the data owner does not have to reveal input or output data to anyone, and at the same time, the confidentiality of the machine learning model is also preserved. In TSM homomorphic encryption is used in the inference context.

B. *Differential Privacy*

Differential Privacy [15], [16] is a formalism for quantifying the degree to which the privacy of each instance of a dataset is preserved when the output of a data analysis algorithm is released. DP guarantees that based on the presence or absence of an instance's data in the dataset, an attacker is unable to draw conclusions about individual instances from the released output of an analysis algorithm. This guarantee is achieved by randomizing the data analysis process. In the machine learning context, randomization is accomplished either by adding random noise to the input or output of the machine learning algorithm, or by modifying the learning algorithm itself. The datasets may contain sensitive information that needs to be protected from so-called model inversion [17], [18] and membership inference [19] attacks, among others. This goal is addressed in the context of DP [20], [21]. The classical approach to approximating a real-valued function with a DP mechanism is to add noise calibrated to the global sensitivity of the function to the function output [22]. However, this introduction of noise into an algorithm generally results in a loss of accuracy. Since DP is preserved in any further processing of the released output data, accuracy can be increased by 'denoising' the output using statistical estimation theory [23]. The iterative nature of machine learning algorithms poses another challenge, as iterations result in a high cumulative loss of privacy and therefore a large amount of noise must be added to compensate for the loss of privacy. To keep track of the loss of privacy arising from successive iterations, Abadi et al. [20] propose a moment accounting method for compositional analysis. The moment-accounting method is based on the properties of privacy loss random variables. Because the method provides a tight bound on the privacy cost of multiple iterations, allowing for a higher privacy budget per iteration, it has been successfully applied to confidentiality-preserving variational Bayesian methods [24]. Distributed deep learning [25], [26], based on distributed stochastic gradient descent (SGDs) methods, is a solution approach to preserve the privacy of each participant and yet learn from the private data of other participants.

A number of libraries are available for practical DP-supported implementations: On the one hand, using libraries such as Google's DP library or the PyDP based on it for direct computations of ε- or (ε, δ)-DP statistics on datasets, and on the other hand, using the machine learning focused libraries TensorFlow Privacy and Opacus , which provide DP-SGD.

If a machine operator offers to train a machine learning model based on training data it has collected, it must anticipate that attempts could be made to extract the original training data from the final model. DP methods provide protection against such extraction attempts and should therefore be used when mission-critical data is used to build machine learning models that are subsequently made available to other parties. In TSM, DP is used in the context of training models that are published following training.

C. *Federated Learning*

The creation of machine learning models is computationally intensive and requires large amounts of memory. To enable the creation of increasingly complex models on ever larger datasets, models and data are fragmented to distribute the training process across multiple devices [27]. However, distributed learning also requires a central coordinator with full access to all data and all devices. If the goal is the creation of shared models by multiple participants, supported by public and internal information of each participant, distributed learning is not a viable approach because the private data cannot be shared.

FL [28] is an extension of distributed learning that addresses this problem and is characterized by three features [29]:

- Massive distribution: no single node has a significant portion of the data.
- Non-IID data: The data cannot be assumed to have an identical and independent distribution (IID) across all participating nodes.
- Unbalanced data: the amount of data may differ by orders of magnitude between any two nodes.

Jakub Konečný et al. [29] have shown that with appropriate federated optimization algorithms that only convey model update information, similar performance can be achieved while keeping the training data private.

In the FL context, since all participants are aware of the global model, malicious participants can also extract information about other participants' training data from the global model, e.g., using model inversion [17], [18] and membership inference attacks [19]. Protection against these threat scenarios can be provided by the use of DP methods [30].

From model updates submitted by participants to the coordinator, e.g., in the form of batch gradients, a malicious coordinator could attempt to reconstruct training data from the batch of the respective participant. Effective protection against this is provided by secure distributed aggregation of model updates using secure multi-party computation tools [31].

Recent work on FL often focuses on the technological aspects of efficient cross-device learning scenarios with edge devices such as smartphones that can drop out of training sessions at any time, while an essential requirement is the need for privacy [32]. Research exists on organizational issues such as shared model ownership through distributed ledgers [33] and malicious attacks or addressing data quality issues in distributed collaborative learning, but these are underrepresented [34]. There is a lack of real-world experiments, as most studies are conducted on simulated use cases.

Meanwhile, a variety of FL libraries also exist, with TensorFlow Federated , FedML , FATE , Intel's OpenFL , and IBM's FL library among the best known.

FL provides a suitable framework for the decentralized creation of machine learning models: For example, multiple machine operators can collaboratively train a machine learning model into which each individual machine operator can incorporate its own machine data, but without having to directly hand it over. This can be coordinated either by a third party (e.g., the machine manufacturer or its own insight provider) or by the group of machine operators themselves.

D. Secure Multi-party Computation

Secure Multi-party Computation is an area of cryptography and information security that allows multiple parties to perform a computation together without revealing their confidential data to each other [35]. The main goal of SMC is to provide a secure way to process private information from multiple parties without one party knowing or making available the other's data.

In traditional computational environments, collaboration among multiple parties would normally require the exchange of data, which could compromise privacy. However, with SMC, these parties can securely cooperate by using a cryptographic protocol structure to share their data and jointly perform a desired computation without the actual raw data being visible to the other parties [36].

The basic idea behind SMC is that each participant retains its own private data, but still receives information that reflects the result of the joint computation. This is done in a way that ensures that even if some parties are malicious or try to obtain information they should not know, the privacy of the other parties is preserved [35].

Secure Multi-party Computation has many practical applications, especially in areas where privacy and confidentiality are important. Examples include [37]:

- Data analytics privacy: companies or organizations can combine and analyze their data to gain insights and trends without disclosing or sharing their sensitive data.
- Secure auctions: Auctions can be held where bidders' bids remain encrypted, but the winning bid can still be determined without knowing the actual bids.
- Secret ballots: Individuals may conduct secret ballots where the results are revealed without the individual votes being known.

To assist in the implementation of SMC, a lot of libraries exist. Some of the better known are ABY, CrypTen, JIFF, Moose, MP-SPDZ and Sharemind. A list of current frameworks has been compiled by Rotaru [38].

III. Evaluation Method

To find which technology is suitable for a particular application, the use case must first be defined. It is helpful to record the use case in writing and to visualise it. When recording, it is important to ensure that all stakeholders, both internal and external, are included. It is advisable to record the use cases along the existing data flows so that no stakeholder is forgotten. It is also important to consider the interests of each stakeholder in the specific use case and where there are concerns about data or information that needs to be protected. This may be process or machine data, but it may also be the machine learning model itself that is deemed worthy of protection. For example, a machine manufacturer may also be a supplier of wear models. In this example, the users of the machine are the users of the model. The intention of the supplier is to rent out his model for profit without giving the users enough information to reproduce the model.

Once the use case has been clearly and comprehensively described, and all stakeholders and their respective interests and security concerns have been noted, the technology selection can begin. In order to support technology selection without requiring expert knowledge of individual technologies, a series of simple yes or no questions were developed to guide the process. Two different methods were chosen to make the selection process clearer. The first method, shown in figure 1, is written from the perspective of a user who has data and wants to use it in a production environment. The second method,

shown in figure 2, approaches the subject from the point of view of a model provider who wants to examine how the machine learning model can be marketed to potential users.

A. *Evaluation for Operators*

about machinery and equipment, it must first be clarified what the critical components of the equipment are. Once this is known, the next step is to determine whether a good enough model already exists to process the machine data. Answering this questions diverge the methods into the HE and the FL track.

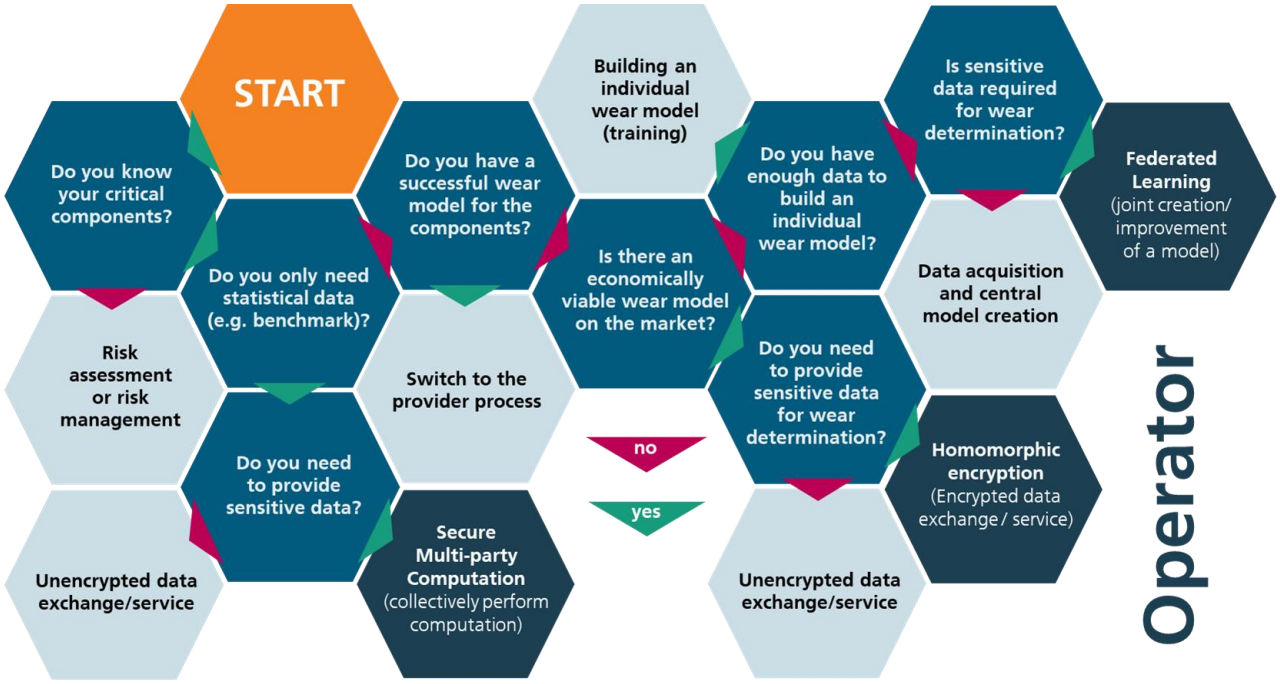

Fig. 1 Privacy preserving technology evaluation method for operators

If a production worker is unsure about which technology to use to analyse existing data, the decision process shown in figure 1 can be used. Simple yes/no questions are used to clarify whether it is known which components are involved and whether simple statistical data is required or whether the existing data needs to be processed using a more complex model. If only simple data is needed and confidential transmission is required, then SMC is suitable for the use case. If more complex models are required, the first step is to determine whether possible models already exist. If so, these can be used with the help of HE, if necessary, so that no confidential data needs to be disclosed. If no model exists, a model can be created using FL. This can be coordinated either by the operator or, as in most cases, in cooperation with the component manufacturer who coordinates the FL. Each point in the decision methodology describes how to proceed even if the conditions of the current use case are not suitable for one of the technologies. These range from the recommendation to first identify the critical components using risk management methods, to the recommendation to build a model yourself, either with your own data or with externally sourced data, or to simply communicating the data unencrypted to a model owner if the level of needed confidentiality is not high.

B. *Evaluation for model providers*

From the point of view of the manufacturer or distributor of a machine learning model, different technologies can be used to maintain confidentiality. In order to identify information

If a model already exists, it must be assessed whether the data used to create the model or the model itself is worthy of protection. If the original data used to create the model is worthy of protection, DP can be used to assess the risk of the original data becoming public either through the use of the model or through targeted attacks. In addition, the supplier of the model needs to consider whether the model can be shared with the customer or whether the model should be protected itself. If critical customer data is also processed in the model, HE is a good choice to protect both the model and the customer data.

If a suitable machine learning model does not yet exist, it either must be created using the own data or, if this is not possible, a model must be created with the help of collaborators. If there is not enough data to create an own model, it must be checked with possible cooperation partners whether the raw data can be collected to create a model or whether the required data is private. If the data needs to be protected, FL is an option to jointly create a model.

C. *Additional criteria to be considered*

In addition to the general selection of the privacy preserving technology, other criteria and restrictions must be considered in each individual use case. The most important selection criteria are summarized so that they can be considered by a production worker without experience in this area of technology.

Most preservation technologies require that participants have *sufficient computing power and communication capacity* to participate efficiently in the learning or data exchange process. Both the computation of model updates and the exchange of these updates between participants, as well as the encryption and decryption in HE, can be time consuming, especially in the case of large data sets or slow network access.

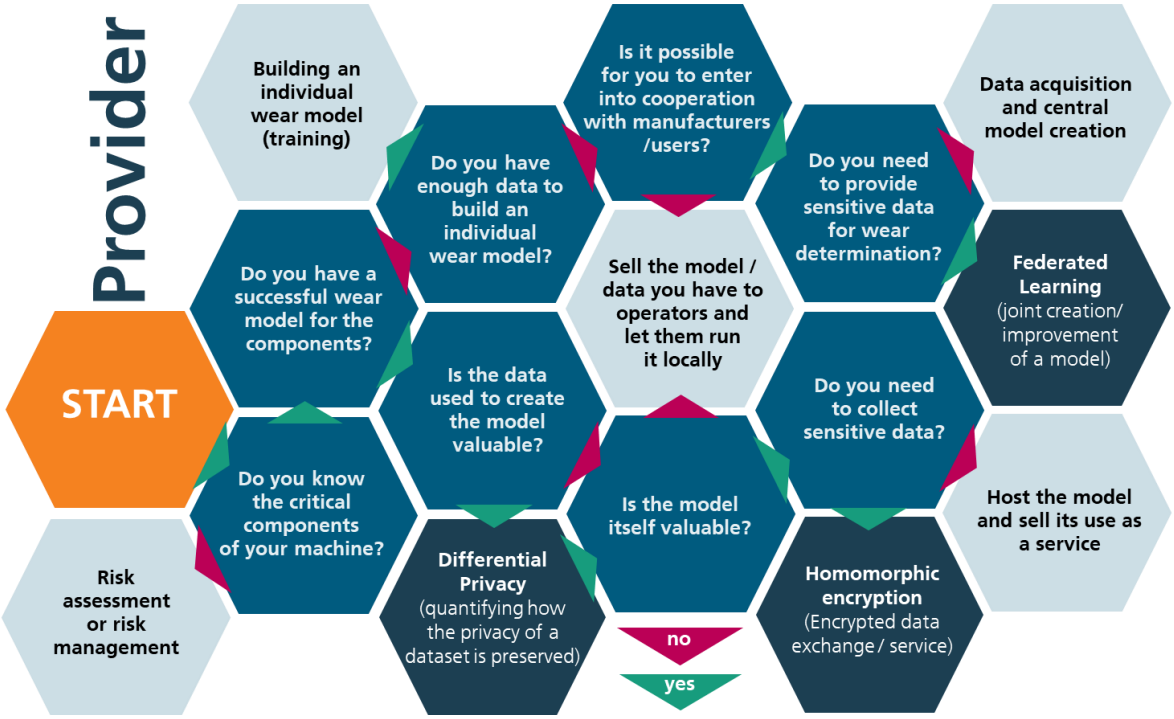

Fig. 2 Privacy preserving technology evaluation method for model provider

It is important to note that *not all data in the cross-enterprise communications needs to be encrypted*. Often both critical process and machine data, which must not be made public, and general data, such as the current temperature in the production hall, flow into machine learning models. In many cases, processing time and hardware requirements can be reduced by including only the critical data with the appropriate technology.

Learning models require that the participants have *sufficient and relevant data* for the model to be effectively trained. If the data within the participant population is highly variable or very limited, this can affect the performance of the model. If the data submitted is encrypted, as is the case with HE, the model cannot be further trained. If this is a requirement for the use case, HE cannot be used.

Some privacy preserving technologies, notably SMC and HE, support only *a limited number of mathematical operations*. Some operations may be inefficient or not supported at all, limiting their use in certain machine learning applications. This should be taken into account when choosing which technique to use. If a new machine learning model is being created, this can be taken into account when creating the model to respond to possible future coding requirements.

IV. Validation

The two evaluation methods presented were discussed with experts from the production environment and validated using different use cases.

In one use case, a wear model was needed for a critical component in order to better plan expensive and time-consuming maintenance. However, the information required to predict wear was proprietary and could not be shared with third parties. For this use case, the *Operator* evaluation method was used. Using this method, the employee deduced on his own, without any further information, that HE was a suitable technology for his use case.

In another use case, a machine manufacturer wanted to develop a wear model, but did not have enough data to develop the model on his own. The users of his machines did not want to share the necessary data. In this case too, the machine manufacturer's employee was able to use the *Provider* evaluation method to identify FL as a suitable technology for his use case without any further information.

V. Conclusions and Discussion

After an overview of the current state of the art in privacy-preserving technologies, two methods were presented that allow an user and a provider without expertise to easily make a pre-selection of the appropriate technology. Based on this, the most important criteria for further selection were described. The validation of the methods with experts has shown that the aim of providing simple methods that can be used by people outside the field has been achieved. However, the discussions also showed that this decision support tool can only be a start and that there is still much to be done in the area of usability and user-friendliness of privacy preserving technologies.

ACKNOWLEDGMENT

We acknowledge support by the Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V. within the project PACT TSM and by the Austrian Research Promotion Agency FFG within the projects SMiLe (4109009) and champI4.0ns (4328338).

REFERENCES

[1] L. Sun, H. Zhang, and C. Fang, "Data security governance in the era of big data: status, challenges, and prospects," *Data Science and Management*, vol. 2, pp. 41–44, 2021, doi: 10.1016/j.dsm.2021.06.001.

[2] W. Zellinger *et al.*, "Beyond federated learning: On confidentiality-critical machine learning applications in industry," *Procedia Computer Science*, vol. 180, pp. 734–743, 2021, doi: 10.1016/j.procs.2021.01.296.

[3] R. Xu, N. Baracaldo, and J. Joshi, "Privacy-Preserving Machine Learning: Methods, Challenges and Directions," 2021, doi: 10.48550/arXiv.2108.04417.

[4] T. Zonta, C. A. Da Costa, R. Da Rosa Righi, M. J. de Lima, E. S. Da Trindade, and G. P. Li, "Predictive maintenance in the Industry 4.0: A systematic literature review," *Computers & Industrial Engineering*, vol. 150, p. 106889, 2020, doi: 10.1016/j.cie.2020.106889.

[5] Y. Ao and Y. Jiang, "Manufacturing Data Privacy Protection System for Secure Predictive Maintenance," in *2022 5th International Conference on Data Science and Information Technology (DSIT)*, Shanghai, China, 2022, pp. 1–5, doi: 10.1109/DSIT55514.2022.9943852.

[6] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, Bethesda MD USA, M. Mitzenmacher, Ed., 2009, pp. 169–178, doi: 10.1145/1536414.1536440.

[7] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) Fully Homomorphic Encryption without Bootstrapping," *ACM Trans. Comput. Theory*, vol. 6, no. 3, pp. 1–36, 2014, doi: 10.1145/2633600.

[8] J. Fan and F. Vercauteren. "Somewhat practical fully homomorphic encryption." https://ia.cr/2012/144 (accessed Jul. 31, 2023).

[9] C. Gentry, S. Halevi, and N. P. Smart, "Better Bootstrapping in Fully Homomorphic Encryption," in *Public Key Cryptography – PKC 2012* (Lecture Notes in Computer Science), D. Hutchison et al., Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 1–16.

[10] C. Gentry, S. Halevi, and N. P. Smart, "Fully Homomorphic Encryption with Polylog Overhead," in *Advances in Cryptology – EUROCRYPT 2012* (Lecture Notes in Computer Science), D. Hutchison et al., Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 465–482.

[11] C. Gentry, A. Sahai, and B. Waters, "Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based," in *Advances in Cryptology – CRYPTO 2013* (Lecture Notes in Computer Science), R. Canetti and J. A. Garay, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 75–92.

[12] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic Encryption for Arithmetic of Approximate Numbers," in *Advances in Cryptology – ASIACRYPT 2017* (Lecture Notes in Computer Science), T. Takagi and T. Peyrin, Eds., Cham: Springer International Publishing, 2017, pp. 409–437.

[13] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song, "Bootstrapping for Approximate Homomorphic Encryption," in *Advances in Cryptology – EUROCRYPT 2018* (Lecture Notes in Computer Science), J. B. Nielsen and V. Rijmen, Eds., Cham: Springer International Publishing, 2018, pp. 360–384.

[14] I. Chillotti, M. Joye, and P. Paillier, "Programmable Bootstrapping Enables Efficient Homomorphic Inference of Deep Neural Networks," in *Cyber Security Cryptography and Machine Learning* (Lecture Notes in Computer Science), S. Dolev, O. Margalit, B. Pinkas, and A. Schwarzmann, Eds., Cham: Springer International Publishing, 2021, pp. 1–19.

[15] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our Data, Ourselves: Privacy Via Distributed Noise Generation," in *Advances in Cryptology - EUROCRYPT 2006* (Lecture Notes in Computer Science), S. Vaudenay, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 486–503.

[16] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *FNT in Theoretical Computer Science*, vol. 9, 3-4, pp. 211–407, 2013, doi: 10.1561/0400000042.

[17] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing," *Proceedings of the ... USENIX Security Symposium. UNIX Security Symposium*, vol. 2014, pp. 17–32, 2014.

[18] M. Fredrikson, S. Jha, and T. Ristenpart, "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, Denver Colorado USA, I. Ray, N. Li, and C. Kruegel, Eds., 2015, pp. 1322–1333, doi: 10.1145/2810103.2813677.

[19] N. Li, W. Qardaji, D. Su, Y. Wu, and W. Yang, "Membership privacy," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security - CCS '13*, Berlin, Germany, A.-R. Sadeghi, V. Gligor, and M. Yung, Eds., 2013, pp. 889–900, doi: 10.1145/2508859.2516686.

[20] M. Abadi *et al.*, "Deep Learning with Differential Privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna Austria, E. Weippl, S. Katzenbeisser, C. Kruegel, A. Myers, and S. Halevi, Eds., 2016, pp. 308–318, doi: 10.1145/2976749.2978318.

[21] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential Privacy Preservation for Deep Auto-Encoders: an Application of Human Behavior Prediction," *AAAI*, vol. 30, no. 1, 2016, doi: 10.1609/aaai.v30i1.10165.

[22] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in *Theory of Cryptography* (Lecture Notes in Computer Science), D. Hutchison et al., Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[23] B. Balle and Y.-X. Wang, "Improving the Gaussian Mechanism for Differential Privacy: Analytical Calibration and Optimal Denoising," 2018, doi: 10.48550/arXiv.1805.06530.

[24] M. Park, J. Foulds, K. Chaudhuri, and M. Welling, "Variational Bayes In Private Settings (VIPS)," *jair*, vol. 68, pp. 109–157, 2020, doi: 10.1613/jair.1.11763.

[25] N. Agarwal, A. T. Suresh, F. Yu, S. Kumar, and H. B. McMahan, "cpSGD: Communication-efficient and differentially-private distributed SGD," 2018, doi: 10.48550/arXiv.1805.10559.

[26] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, 2015, pp. 909–910, doi: 10.1109/ALLERTON.2015.7447103.

[27] D. Peteiro-Barral and B. Guijarro-Berdiñas, "A survey of methods for distributed machine learning," *Prog Artif Intell*, vol. 2, no. 1, pp. 1–11, 2013, doi: 10.1007/s13748-012-0035-5.

[28] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," 2016, doi: 10.48550/arXiv.1602.05629.

[29] J. Konečný, B. McMahan, and D. Ramage, "Federated Optimization:Distributed Optimization Beyond the Datacenter," 2015, doi: 10.48550/arXiv.1511.03575.

[30] K. Wei *et al.*, "Federated Learning With Differential Privacy: Algorithms and Performance Analysis," *IEEE Trans.Inform.Forensic Secur.*, vol. 15, pp. 3454–3469, 2020, doi: 10.1109/TIFS.2020.2988575.

[31] K. Bonawitz *et al.*, "Practical Secure Aggregation for Privacy-Preserving Machine Learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Dallas Texas USA, B. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds., 2017, pp. 1175–1191, doi: 10.1145/3133956.3133982.

[32] S. K. Lo, Q. Lu, C. Wang, H.-Y. Paik, and L. Zhu, "A Systematic Literature Review on Federated Machine Learning," *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–39, 2022, doi: 10.1145/3450288.

[33] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained On-Device Federated Learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, 2020, doi: 10.1109/LCOMM.2019.2921755.

[34] Z. Gu *et al.*, "Reaching Data Confidentiality and Model Accountability on the CalTrain," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Portland, OR, USA, 2019, pp. 336–348, doi: 10.1109/DSN.2019.00044.

[35] D. Evans, V. Kolesnikov, and M. Rosulek, "A Pragmatic Introduction to Secure Multi-Party Computation," *FNT in Privacy and Security*, vol. 2, 2-3, pp. 70–246, 2018, doi: 10.1561/3300000019.

[36] A. B. Alexandru and G. J. Pappas, "Secure Multi-party Computation for Cloud-Based Control," in *Privacy in Dynamical Systems*, F. Farokhi, Ed., Singapore: Springer Singapore, 2020, pp. 179–207.

[37] W. Du and M. J. Atallah, "Secure multi-party computation problems and their applications," in *Proceedings of the 2001 workshop on New security paradigms*, Cloudcroft New Mexico, V. Raskin, S. J. Greenwald, B. Timmerman, and D. Kienzle, Eds., 2001, pp. 13–22, doi: 10.1145/508171.508174.

[38] D. Rotaru. "awesome-mpc." https://github.com/rdragos/awesome-mpc #frameworks (accessed Jul. 31, 2023).