3rd Conference on Production Systems and Logistics

Local Differential Privacy In Smart Manufacturing: Application Scenario, Mechanisms and Tools

Sascha Gärtner[1], Michael Oberle[1]
[1]Fraunhofer IPA, Stuttgart, Germany

Abstract

To utilize the potential of machine and deep learning, enormous amounts of data are required. A common and beneficial approach is to share datasets between the parties involved for training purposes or even to release datasets to the public. However, several incidents have shown that despite anonymizing the data, attackers are still capable of identifying individuals in the data and extracting their sensitive information. The methods of differential privacy address this problem by adding a statistical noise to data points in the shared dataset. Since manufacturing data not only contains information about individual persons but also about the companies, their process knowledge, products, and orders add more complexity to the application of differential privacy compared to other domains. In this paper, we highlight why conventional methods of anonymization are not sufficient to guarantee data protection and thus present the necessity of using differential privacy. To illustrate its usefulness for manufacturing we present a specifc application scenario and examine potential threats when sharing manufacturing data. We identify mechanisms to perturbate data and map these to variable types in the manufacturing context. To guide practical application and research we finally outline existing differntial privacy libraries, and highlight current limitations.

Keywords

Local differential privacy; smart manufacturing, industrial Internet of Things, privacy preservation, LDP;

1. Introduction

Driven by the digital transformation, traditional manufacturing is currently undergoing a change towards smart manufacturing. The digital transformation is especially initiated by key technologies such as the Internet of Things, 5G, CPSs, BigData, and Artificial Intelligence (AI) [1,2]. Due to the application of sensor technology and IT infrastructure, the amount of data generated during manufacturing processes are constantly increasing [3]. However, the amounts of data are often not sufficient to train generalizable machine learning or AI models. To address this issue, approaches such as the establishment of common data spaces and federated learning for collaborative training of algorithms are emerging [4]. Numerous real-world examples [5–7] have shown that sharing data with third parties is critical in terms of privacy violations. Typically, privacy is violated when attackers identify individuals in the published dataset and thus have access to sensitive information [8]. In the context of manufacturing, privacy threats are much more complex and increase since additional sensitive company-relevant data can be identified. For this reason, we contribute by mapping the concepts of differential privacy to the manufacturing context and presenting a real-world application scenario including perturbation mechanisms. At the beginning of the paper, we give an overview of the general motivations of differential privacy by comparing conventional methods and mentioning well-known privacy leaks. We then map the concept of differential privacy to manufacturing and specify the problem to machine manufacturers and their customers who operate the machines in their

DOI: https://doi.org/10.15488/12125 publish-Ing.

factories. We then identify relevant parameters that occur in manufacturing, categorize them by variable type, and present examples of suitable differentially private mechanisms. Finally, we provide an overview of libraries and toolboxes for guiding the practical application of differential privacy, outline topics for further research and conclude the paper by highlighting the key points.

1.1 Privacy concerns and the need for differential privacy

Typical techniques to ensure the privacy of user data are masking, generalization, and k-anonymization. For additional security, these techniques can be complemented by encryption mechanisms such as homormorphic encryption [9]. However, the use of these techniques is vulnerable to a variety of attacks (e.g., linkage, reconstruction, and differentiation attacks). Linkage attacks for example use similar publicly available datasets to find similarities within the data. It has been proven that even a few data points are sufficient to uniquely identify individual persons [10,7]. The encryption of individual sensitive data points in a data series also involves vulnerabilities. If the attacker succeeds in gaining knowledge of the function used for encryption, the data can be decrypted again by systematically testing possible input values [11,12].

To overcome these problems, the research field differential privacy emerged. Differential privacy can be seen as a process A, applied to some data D. The process might be the estimation of the mean over the distribution of a dataset or a machine learning process to predict values. To achieve the formal definition of differential privacy the process A has to be modified. This is usually done by adding noise at a certain point in the process. Adding the right amount of noise strongly depends on the use case and threat model.

Considering two neighboring datasets $D_1$and D_2, where dataset D_2 differs from dataset D_1 by just a record of a person, the process A is considered ε-differentially private if the output O of the process is approximately the same when being applied to both datasets. This leads to approximately identical probabilities $\mathbb{P}$. The relationship between the two probabilities is described by the following definition [13].

$$\mathbb{P}[A(D_1) = O] \leq e^{\varepsilon} \cdot \mathbb{P}[A(D_2) = O] \qquad (1)$$

The mechanism used to add noise is dependent on the data type. Typically, the Randomized Response, Laplace, Gaussian, and Exponential mechanisms are used. The parameterization must be adapted in each case to the variable to be determined. Several real-world applications demonstrate the potential, but also the complexity, of differential privacy. Apple uses differential privacy to collect data from end-users of iOS or macOS [14,15]. For example, words that are typed by a sufficient number of users but are not yet in the dictionary are collected differentially private. Facebook created and released a dataset that provides information about user interactions with websites that have been shared on their platform [16].

2. Scenario of differential privacy in manufacturing

While the purpose of Differential privacy is easily accessible when exposing data to the general public to protect the privacy of individuals, the transfer of use cases to manufacturing is not immediately apparent. Considering the paradigm shift from traditional production to autonomous manufacturing, the relevance of data-driven approaches to make manufacturing processes more efficient is constantly increasing. The importance of data for the optimization of processes, plants, and machines is accordingly high. Sharing company-related or process data in manufacturing is therefore unavoidable for companies. [17]

During our research, we identified a common scenario that represents the current issues and concerns of manufacturing companies in merging and sharing data for training machine learning algorithms. There is a trend to improve the customer's process by offering value-added services additional to the machine itself, thus opening up new business areas [18]. The machine manufacturer (curator), wants to collect data from the machines in the customer's productive operation in order to subsequently optimize the machine. The motivations can be constructive improvements of the machine through insights into the daily production operation, quality checks or improved control loops of the machine. [19]

Regardless of whether the machine manufacturer wants to process the aggregated data of the customer with statistical methods, machine learning, or artificial intelligence, there are two possible ways (Figure 1) for the customer to share his data. The differential private mechanism $M$is either held by the curator (GDP-Model) or by the customer (LDP).[20]

In the global differential privacy (GDP) model, the customer can share his raw data with the curator. In this case, the customer has to fully trust the curator. It is not necessarily defined how and whether the curator provides the data to third parties or other customers. The curator can aggregate the datasets of individual customers and thus host a dataset in total. Other customers or external parties can make requests to learn distributions of certain quantities in the dataset. The privacy of the customer data can be protected if the output of the query is appropriately noisy.

In the local differential privacy (LDP) model, noise is added to raw data before sending it to the curator. This model has the advantage that the curator does not have to be trusted. It should be noted that data can be of any variable type. The concept of federated learning enables distributed learning of a shared neural network [4]. In this case, the curator only aggregates the weights of the model. Prior to this, the client trains the model on a local instance (e.g. edge-device). Privacy is achieved by adding noise to the gradients, objective, or output during the training of the model. The biggest advantage of federated learning is that the mechanism works reliably regardless of the data type. However, there are several disadvantages. The customer must have the computing resources to perform the training of the neural network on the edge. In addition, qualified specialists are needed to implement the necessary IT infrastructure and pipelines [21]. In consequence this leads to high costs related to setup and operation. If the purpose and benefits are not immediately apparent to the customer, skepticism arises and they are not willing to make investments. For this reason, we refer to a scenario in which the customer has no computing resources locally available.

A more effective solution in terms of cost and effort is offered by adding noise directly to the raw data. Due to many different types of parameters and attributes occurring on the shopfloor in manufacturing, adding noise to raw data is more complex and must be adapted to the individual case. To select suitable mechanisms and therefore ensure privacy protection it is necessary to identify type of the variable first.

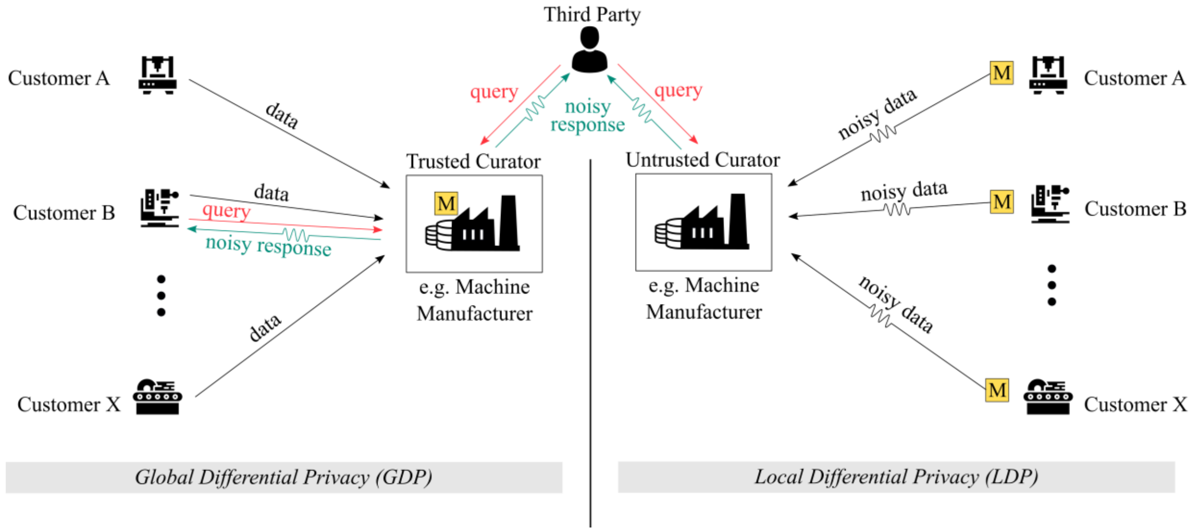


Figure 1: Comparison between global and local differential privacy

3. Parameters in manufacturing and corresponding threats

In order to apply differential privacy to use cases in the manufacturing context, we first determine parameters that occur in the manufacturing environment. The parameters can be divided into different categories and specified (see Table 1). Each parameter category represents different types of information. Consequently, there are different risks in the case of a violation of privacy. While the manufacturing process parameters describe the manufacturing process itself, the process parameters further specify the process with its attributes. These parameters contain the manufacturer's main process knowledge for creating the product. The environment condition parameters have an indirect influence on the process. The ambient temperature can influence the thermal expansion of components within the machine and the workpiece itself, which in turn affects the quality of the workpiece. The attributes listed below can be of different variable types. For the application of mechanisms to satisfy differential privacy, the variable type plays a decisive role.

Table 1: Different kinds of sensitive parameters occurring in the manufacturing context according to [22]

Parameter Category	Attribute
Manufacturing Process Parameters	milling, turning, laser cutting, welding, casting, extrusion, stamping, assembling, etc.
Process Parameters	spindle speed, cutting speed, pressure, coolant, voltage, feed, current, force, torque, etc.
Environment Condition Parameters	humidity, temperature, date, time, rainfall, etc.
Working Condition Parameters	duration, shift, worker id, machine id, etc.
Target Parameters	quality, yield, productivity, OEE, KPIs, etc.
Other Parameters	manufacturing order, material, production numbers, geometric data, position, etc.

If we assume a company shares a large dataset containing the variables listed above multiple threats about the company's process knowledge as well as sensitive business information can arise which are not immediately apparent. A potential attacker can analyze environment condition parameters in the dataset and combine them with publicly available weather data. If there are sufficient matches over time it is possible to identify the company itself at its location. Together with other parameters such as manufacturing orders, date, material, supply chains, or even suppliers thus the order situation might be revealed. If the data is then complemented with target parameters, attackers can reveal the company's productivity and efficiency in production. Therefore, it would be possible to estimate the turnover or profit of a company.

Corporate-related threats link to the process knowledge, which is required to produce workpieces efficiently while ensuring quality. Potential attackers can use the type of the manufacturing process, machine ID, and the associated process parameters to gain sensitive process knowledge about the setup of the machine during the ramp-up process. Another threat can occur if the position data of AGVs is regarded in conjunction with the time stamp. This allows attackers to determine the factory layout and retrace the production routes within the factory. The previously mentioned threats should not be regarded as comprehensive. These are only a few examples for illustrating the problems that can arise from exposing the mentioned variables.

4. Data perturbation mechanisms

Different mechanisms can be used to prevent the exposure of the previously listed threats by adding noise to data before sharing. However, the mechanism used is directly dependent on the variable type and its parameterization is not trivial. In the following, the most common variable types are explained and their context for manufacturing is presented. In addition, the mechanism by which differential privacy can be satisfied and the analogies to related areas are described.

Essentially, the Randomized Response, Laplace Mechanism, Gaussian Mechanism, or Exponential Mechanism are used to add noise [23,24]. Depending on the application case the functions are parameterized by the so-called sensitivity. Hereby the privacy budget from low to high can be adjusted.

Most publications and thus algorithms refer to the assumption that a user owns a data set and a potential attacker issues a query (e.g. mean value of a numeric variable) to the user's database. The query can be connected by AND, OR conditions [25]. In this case, the noise will be applied to the true value and then sent to the potential attacker. Depending on the sensitivity, the database size, and the number of variables, the number of possible queries of a user has to be limited. In addition, the number of combinations of AND, OR conditions also has to be limited. Since the determination of the noise, as well as the boundary conditions to the queries, must be individually adjusted for each problem, the implementation is very extensive.

Our desired approach is an algorithm that allows publishing the dataset under conditions of differential privacy instead of hosting a dataset and answering the queries of clients. The goal is to apply noise to a dataset so that it can be passed on to third parties without being concerned about query limitations. Sensitive information or process variables can not be identified in this case. But the data should still contain enough information to learn the statistical distribution of the variable. For example, it would still be possible to learn something about the wear of the machine and upcoming maintenance but it is not possible to determine the exact process parameters used for manufacturing the workpieces. In reality, many different and complex data structures exist. In the following, we present suitable mechanisms for each variable type and highlight the relation to manufacturing.

4.1 Binary data

The most straightforward approach to publishing a variable differentially private is for binary variables. A very effective approach is the so-called Randomized Response.[26] As an example, the question is asked whether the machine owner (user) has used coolant in a certain process section. The possible answers in this scenario are just *yes=1 or no=0*. Before answering the question, an imaginary coin is tossed. If the coin lands on heads, the user must answer truthfully. If the coin lands on tails, a second coin is tossed. If the second coin lands on heads the user answer with *yes*, otherwise with *no*. This algorithm is differentially private by definition. Therefore, the user can safely reveal the truth. Though noise is also being added to the data by the mechanism. However, if a large set of responses from different companies is received, the noise can be canceled out and the statistical distribution of the use of coolant in the process can be determined. However, it cannot be determined whether a specific company used coolant in its process or not. The most known application of this algorithm is called RAPPOR (extended with additional operations). It was developed and used by Google to determine the default search engines of Google Chrome users [27].

4.2 Numerical data

Applying noise to numerical data can be achieved by using the Laplace or Gaussian mechanism [28]. These mechanisms represent a distribution with probability values and a scaling factor. If we ask a machine user how often the fixture did break during the last year, we expect a numerical value as a response. Before publishing the true value, the dataset holder selects a random value from the Laplace distribution, adds it to the true value, and then writes the perturbed value to the dataset. Instead of the correct number of fixture breakages, a noisy value is given. If multiple machine users report their insert breakage data, the machine manufacturer will be able to determine the average without revealing how often the insert really broke at each user. Other numerical queries could be the calculation of the mean over several data rows or the query about the numerical distribution of a parameter. For example, a querier (machine manufacturer) may ask the question, how often machine failures in the range of [0-9;10-19;20-29;…] occurred. The output would thus be a histogram, which can also be released under the satisfaction of differential privacy. It is possible to apply an individual noise to each count by applying the distribution function to each single count.

4.3 Categorical data

Publishing categorical variables taking into account differential privacy can be seen as an extended version of Randomized Response. But instead of two categories (0 and 1), finite categories are possible. An implementation is the Google algorithm RAPPOR [29]. Other algorithms are the Local Hash method [30] and the Unary Encoding Method [31] which is the basic concept of RAPPOR. The Unary Encoding method (Figure 2) is very intuitive and is presented using an example in the manufacturing context. A querier (machine manufacturer) wants to know which clamping tools his customers use in their production. All parties agree that there are four different possibilities in total. These four possibilities are each represented by a position within a bit string. Position 1 in the bit string stands for the three-jaw chuck, position 2 for the four-jaw chuck, position 3 for the collet chuck, and position 4 for the centering tip. The customer now encodes his clamping tool used in production into the bit string. Then each bit is perturbed according to the Randomized Response method. The perturbed bit strings will then be sent to the querier. The querier adds up the individual positions in the bit string and can thus calculate a distribution of the clamping tools used. However, the querier does not know which exact clamping tool is used by which customer. It should be noted that the accuracy increases significantly with a higher number of contributions.

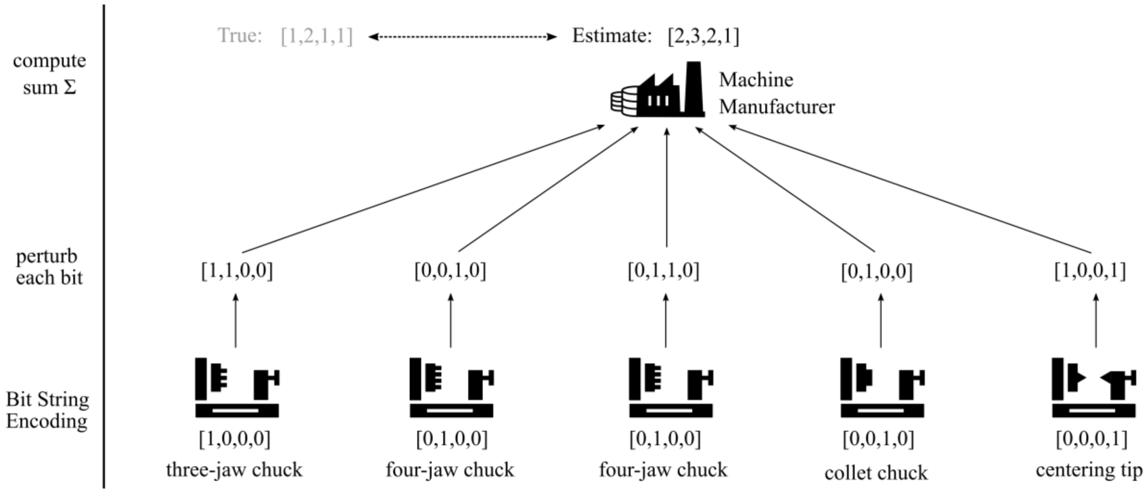


Figure 2: Example of Unary Encoding according to [31]

4.4 Time series data

Time series are becoming increasingly important due to smart sensor technology and advances in data transmission rates. It is possible to reliably record data at high sampling rates, to process and store it outside the PLC. In the context of data analysis for monitoring machines or predicting the occurrence of events, the analysis of time series is of great relevance. In addition to sensor data, time series can also be multidimensional position data from IoT devices on the shopfloor. There is no generally valid way to obtain information from time series. For example, a time series (e.g. sensor signal) can be sampled with sliding windows. A querier might be interested in the average value of each sliding window. Then response would just be a noisy numeric value, as seen above. The analogy to the histogram mentioned before would be if the querier asks for the frequency spectrum of a periodic signal. However, it is also possible to publish several time series under the condition of differential privacy. For example, this can be achieved by a re-quantization mapping [32]. Another reference introduces a perturbation mechanism consisting of several single steps to satisfy differential privacy [33].

Coordinate, position, or trajectory data often differ from one-dimensional time series by the property of multidimensionality. Publishing coordinate data is rarely addressed in the literature. However, some papers show possible ways how this can be achieved. [34–36]

5. Available libraries and tools for implementing differential privacy

In order to apply differential privacy efficiently in manufacturing scenarios, it is necessary to keep the implementation as simple as possible. It should be avoided to implement the algorithms from scratch since it is too error-prone. Libraries with single building blocks and ready-to-use mechanisms are the preferred alternative. There are several libraries available that provide the basic mechanisms for different programming languages (e.g. Python, Java, C, GO, C++). We give a brief overview about some of them.

A library that enables the application of basic mechanisms is the *Google Differential Privacy library* [37]. With this library at the same time, the mechanisms can be built on existing frameworks such as Apache Beam. The *Google RAPPOR library* [29] enables the application of the RAPPOR algorithm which was previously presented in the context of the Randomized Response Method. The *OpenDP* project [38] also provides easy access to the mechanisms to apply them to individual data with the *smartnoise library*. *IBM* offers a library to use the discrete Gaussian mechanism [39]. *Ektelo* is also a framework for implementing privacy algorithms [40].

Furthermore, libraries exist that allow the training of neural networks under conditions of differential privacy. The *Opacus* [41] framework allows differentially private training of PyTorch models. *Opacus* is open source and offers a modular API. *TensorFlow* [42] also offers a library that enables the differential private training of neural networks. The *OpenMined Project* [43], with its *Syft* and *Grid* modules, applies differential privacy in the context of federated learning. The project is compatible with PyTorch and TensorFlow. The *diffprivlib* [44] by *IBM* offers basic mechanisms, which can be applied individually by the user. However, simple machine learning algorithms such as a random forest or a logistic regression can also be trained under differential privacy with the *diffprivlib*.

Since it is not trivial to determine how the algorithms are implemented in detail in each library, a comparison was conducted by Garrdio et.al [45]. The libraries were compared qualitatively, and quantitatively with four different types of queries using synthetic and real-world datasets. All libraries were suitable for productional use, however, they differ strongly in the function range. However, no library satisfies a universal utility for all applications. [45]

6. Directions for future research

The practical examples in section 5 show that specific mechanisms are needed considering different variable types. Extending the mechanisms to publishing multidimensional data, i.e. mixed data containing numeric and categorical data types, is not trivial. Research shows that applying the mechanisms to the individual attributes yields poor results. Therefore, solutions must be developed that can perturb multidimensional datasets in total containing numeric and categorical variables with the optimal worst-case error. [46–48].

The given examples also show that different queriers who act independently externally but combine their knowledge gained later, must be taken into account during the design of a LDP system. This comes into effect if other third parties can have access to the data instead of just the machine manufacturer. In case of doubt, the number of requests from each querier and the number of same requests must be limited [49]. If each analyst receives a slightly different answer, analysts could collaborate and calculate the mean of their answers. In a worst-case scenario, they are able to determine the true value. In this case, it makes sense to limit the number of queries and send the same answer to each analyst [28]. After defining the collaborators and the variable types to be published have been determined, the mechanisms for adding the noise must be suitably parameterized. Since the parameter epsilon ε is a measure of privacy and is also needed for the parameterization of the mechanisms, the choice of this value is very crucial. Currently there is no best practice for setting ε for a desired privacy utility tradeoff. Thus it would be helpful if early adopters of differential privacy could share their ε values from real-world applications [50].

In the long term, it would be desirable to be able to publish entire differentially private datasets [49]. In the context of Open Science, the release of whole datasets would also be advantageous. Companies could share their data with the machine learning community without having any privacy concerns. The use of machine learning and artificial intelligence would become more quickly applicable through the collaborative work of the community and thus take a further step toward autonomous production.

7. Conclusion

By identifying and defining an application scenario, mapping the concepts of LDP to the manufacturing context, we have shown that the demands and potential threats to privacy leaks when publishing or sharing data with third parties are of a different kind compared to the threats when considering public datasets for the protection of individual personal data. It must be understood that the company's process knowledge can be leaked by sharing production data with third parties. Beyond important process information, which is necessary to produce cost-effective products, sensitive business data as well as strategic data can be revealed. To apply LDP in the manufacturing context, it is mandatory to analyze the use case in advance. It should be asked who will have access to what kind of data and which potential threats can arise by sharing the data. From the point of view of the curator (machine manufacturer), it must be taken into account that the data amounts must be correspondingly large in order to learn valid insights. In general, when publishing differentially private data, it must be taken into account that there is a tradeoff between accuracy and privacy. There is no generic approach for determining the ideal value of the parameter yet.

Acknowledgments

This work was partially supported by a grant from the Federal Ministry of Education and Research providing for this work through the project "DigiBattPro 4.0" and from the State Ministry of Baden-Wuerttemberg for Economic Affairs, Labour and Tourism (Project *"KI-Fortschrittszentrum: Lernende Systeme und Kognitive Robotik"*).

References

[1] Arinez, J.F., Chang, Q., Gao, R.X., Xu, C., Zhang, J., 2020. Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook. Journal of Manufacturing Science and Engineering 142 (11).

[2] Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., Ueda, K., 2016. Cyber-physical systems in manufacturing. CIRP Annals 65 (2), 621–641.

[3] Wang, L., 2019. From Intelligence Science to Intelligent Manufacturing. Engineering 5 (4), 615–618.

[4] Savazzi, S., Nicoli, M., Bennis, M., Kianoush, S., Barbieri, L., 2021. Opportunities of Federated Learning in Connected, Cooperative, and Automated Industrial Systems. IEEE Commun. Mag. 59 (2), 16–21.

[5] Arvind Narayanan, Vitaly Shmatikov, 2006. Robust De-anonymization of Large Datasets (How to Break Anonymity of the Netflix Prize Dataset).

[6] M. Douriez, H. Doraiswamy, J. Freire, C. T. Silva, 2016. Anonymizing NYC Taxi Data: Does It Matter?, in: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA). 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 140–148.

[7] Pierangela Samarati, L.S., 1998. Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression.

[8] Liu, K., Giannella, C., Kargupta, H., 2008. A Survey of Attack Techniques on Privacy-Preserving Data Perturbation Methods, in: Aggarwal, C.C., Yu, P.S. (Eds.), Privacy-preserving data mining. Models and algorithms, vol. 34. Springer, New York, NY, pp. 359–381.

[9] Singh, N., Singh, A.K., 2018. Data Privacy Protection Mechanisms in Cloud. Data Sci. Eng. 3 (1), 24–39.

[10] Martin M. Merener, 2012. Theoretical Results on De-Anonymization via Linkage Attacks. Transactions on Data Privacy 5 (2), 377–402.

[11] Naveed, M., Kamara, S., Wright, C.V., 2015. Inference Attacks on Property-Preserving Encrypted Databases, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. CCS'15: The 22nd ACM Conference on Computer and Communications Security, Denver Colorado USA. 12 10 2015 16 10 2015. ACM, New York, NY, pp. 644–655.

[12] Rigaki, M., Garcia, S., 2020. A Survey of Privacy Attacks in Machine Learning. https://arxiv.org/pdf/2007.07646.

[13] Dwork, C., 2006. Differential Privacy, in: Hutchison, D., Kanade, T., et.al. (Eds.), Automata, languages and programming. 33rd international colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006; proceedings, vol. 4052. Springer, Berlin, pp. 1–12.

[14] A. G. Thakurta, A. H. Vyrros, U. S. Vaishampayan, G. Kapoor, J. Freudiger, V. R. Sridhar, and D. Davidson. Learning new Words.

[15] Differential Privacy Team, 2017. Learning With Privacy at Scale.

[16] Messing, S., DeGregorio, C., Hillenbrand, B., King, G., Mahanti, S., Mukerjee, Z., Nayak, C., Persily, N., State, B., Wilkins, A., 2020. Facebook Privacy-Protected Full URLs Data Set.

[17] Braud, A., Fromentoux, G., Radier, B., Le Grand, O., 2021. The Road to European Digital Sovereignty with Gaia-X and IDSA. IEEE Network 35 (2), 4–5.

[18] Siderska, J., Jadaan, K.S., 2018. Cloud manufacturing: a service-oriented manufacturing paradigm. A review paper. Engineering Management in Production and Services 10 (1), 22–31.

[19] Wang, K., 2006. Data Mining in Manufacturing: The Nature and Implications, in: Wang, F., Wang, K., Kovacs, G., Wozny, M., Fang, M. (Eds.), Knowledge enterprise: intelligent strategies in product design, manufacturing, and management. Proceedings of PROLAMAT 2006, IFIP TC5 international conference, June 15-17 2006, Shanghai, China, vol. 207, 1. Ed. ed. Springer, New York, NY, pp. 1–10.

[20] Mahawaga Arachchige, P.C., Bertok, P., Khalil, I., Liu, D., Camtepe, S., Atiquzzaman, M., 2019. Local Differential Privacy for Deep Learning 7. https://arxiv.org/pdf/1908.02997.

[21] D Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Dan Dennison, 2015. Hidden Technical Debt in Machine Learning Systems. Advances in Neural Information Processing Systems, 2494–2502.

[22] Wang, K., 2007. Applying data mining to manufacturing: the nature and implications. J Intell Manuf 18 (4), 487–495.

[23] Aggarwal, C.C., Yu, P.S. (Eds.), 2008. Privacy-preserving data mining: Models and algorithms. Springer, New York, NY, 513 pp.

[24] Hassan, M.U., Rehmani, M.H., Chen, J., 2020. Differential Privacy Techniques for Cyber Physical Systems: A Survey. IEEE Commun. Surv. Tutorials 22 (1), 746–789.

[25] Yang, M., Lyu, L., Zhao, J., Zhu, T., Lam, K.-Y., 2020. Local Differential Privacy and Its Applications: A Comprehensive Survey, 24 pp. https://arxiv.org/pdf/2008.03686.

[26] Warner, S.L., 1965. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. Journal of the American Statistical Association 60 (309), 63.

[27] Erlingsson, Ú., Pihur, V., Korolova, A., 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response, 14 pp. https://arxiv.org/pdf/1407.6981.

[28] Leoni, D., 2012. Non-interactive differential privacy, in: Proceedings of the First International Workshop on Open Data. the First International Workshop, Nantes, France. 5/25/2012 - 5/25/2012. ACM, New York, NY, p. 40.

[29] Google. RAPPOR. https://github.com/google/rappor. Accessed 3 February 2022.

[30] Bassily, R., Smith, A., 2015. Local, Private, Efficient Protocols for Succinct Histograms, in: Proceedings of the forty-seventh annual ACM symposium on Theory of computing. STOC '15: Symposium on Theory of Computing, Portland Oregon USA. 14 06 2015 17 06 2015. ACM, New York, NY, pp. 127–135.

[31] Tianhao Wang, Jeremiah Blocki, and Ninghui Li, Somesh Jha. Locally Differentially Private Protocols for Frequency Estimation, in: , Proceedings of the 26th USENIX Security Symposium, vol. 26.

[32] S. Xiong, A. D. Sarwate, N. B. Mandayam, 2016. Randomized requantization with local differential privacy, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2189–2193.

[33] Ye, Q., Hu, H., Li, N., Meng, X., Zheng, H., Yan, H., 2021. Beyond Value Perturbation: Local Differential Privacy in the Temporal Setting. INFOCOM 2021 - IEEE Conference on Computer Communications, 1–10.

[34] Bi, M., Wang, Y., Cai, Z., Tong, X., 2020. A privacy-preserving mechanism based on local differential privacy in edge computing. China Commun. 17 (9), 50–65.

[35] Jiang, K., Shao, D., Bressan, S., Kister, T., Tan, K.-L., 2013. Publishing trajectories with differential privacy guarantees, in: Proceedings of the 25th International Conference on Scientific and Statistical Database Management. the 25th International Conference, Baltimore, Maryland. 7/29/2013 - 7/31/2013. ACM, New York, NY, p. 1.

[36] Kim, J.W., Kim, D.-H., Jang, B., 2018. Application of Local Differential Privacy to Collection of Indoor Positioning Data. IEEE Access 6, 4276–4286.

[37] Google. Google Differential Privacy. https://github.com/google/differential-privacy. Accessed 3 February 2022.

[38] OpenDP. smartnoise-sdk. https://github.com/opendp/smartnoise-sdk. Accessed 3 February 2022.

[39] IBM. discrete gaussian differential privacy. https://github.com/IBM/discrete-gaussian-differential-privacy. Accessed 3 February 2022.

[40] Ektelo. Ektelo. https://github.com/ektelo/ektelo. Accessed 3 February 2022.

[41] Opacus. Opacus. https://opacus.ai/. Accessed 3 February 2022.

[42] TensorFlow. TensorFlow Privacy. https://github.com/tensorflow/privacy. Accessed 3 February 2022.

[43] OpenMined. PySyft. https://github.com/OpenMined/PySyft. Accessed 3 February 2022.

[44] IBM. differential-privacy-library. https://github.com/IBM/differential-privacy-library. Accessed 3 February 2022.

[45] Garrido, G.M., Near, J., Muhammad, A., He, W., Matzutt, R., Matthes, F., 2021. Do I Get the Privacy I Need? Benchmarking Utility in Differential Privacy Libraries, 13 pp. https://arxiv.org/pdf/2109.10789.

[46] Nguyên, T.T., Xiao, X., Yang, Y., Hui, S.C., Shin, H., Shin, J., 2016. Collecting and Analyzing Data from Smart Device Users with Local Differential Privacy, 11 pp. https://arxiv.org/pdf/1606.05053.

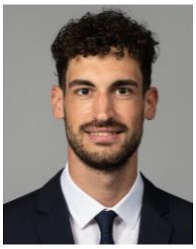
[47] Wang, N., Xiao, X., Yang, Y., Zhao, J., Hui, S.C., Shin, H., Shin, J., Yu, G., 2019. Collecting and Analyzing Multidimensional Data with Local Differential Privacy. https://arxiv.org/pdf/1907.00782.

[48] Wang, T., Ding, B., Zhou, J., Hong, C., Huang, Z., Li, N., Jha, S., 2019. Answering Multi-Dimensional Analytical Queries under Local Differential Privacy, in: Proceedings of the 2019 International Conference on Management of Data. SIGMOD/PODS '19: International Conference on Management of Data, Amsterdam Netherlands. 30 06 2019 05 07 2019. Association for Computing Machinery, New York,NY,United States, pp. 159–176.

[49] Mohammed, N., Chen, R., Fung, B.C., Yu, P.S., 2011. Differentially private data release for data mining, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. the 17th ACM SIGKDD international conference, San Diego, California, USA. 8/21/2011 - 8/24/2011. ACM, New York, NY, p. 493.

[50] Dwork, C., Kohli, N., Mulligan, D., 2019. Differential Privacy in Practice: Expose your Epsilons! JPC 9 (2).

Biography



Sascha Gärtner (*1993), is a research associate at the Fraunhofer Institute for Manufacturing Engineering and Automation (IPA) in Stuttgart, Germany. As part of his work in the Competence Center for Digital Tools in Production, he researches the practical application of artificial intelligence methods in the smart manufacturing environment.

Michael Oberle (*1984) is a computer scientist and group leader working in the field of smart manufacturing with a focus of data-driven and event- driven production control services. One of his main achievements is the fully connected cloud-controlled battery manufacturing pilot line at Fraunhofer IPAs technical centre for battery manufacturing.